International Journal of Information, Control and Computer Sciences
ISSN: 2517-9942
Vol:1, No:7, 2007

A Cheating Model for Cellular Automata-Based Secret Sharing Schemes

Borna Jafarpour, Azadeh Nematzadeh, Vahid Kazempour, and Babak Sadeghian

***Abstract*—**Cellular automata have been used for design of cryptosystems. Recently some secret sharing schemes based on linear memory cellular automata have been introduced which are used for both text and image. In this paper, we illustrate that these secret sharing schemes are vulnerable to dishonest participants' collusion. We propose a cheating model for the secret sharing schemes based on linear memory cellular automata. For this purpose we present a novel uniform model for representation of all secret sharing schemes based on cellular automata. Participants can cheat by means of sending bogus shares or bogus transition rules. Cheaters can cooperate to corrupt a shared secret and compute a cheating value added to it. Honest participants are not aware of cheating and suppose the incorrect secret as the valid one. We prove that cheaters can recover valid secret by removing the cheating value form the corrupted secret. We provide methods of calculating the cheating value.

***Keywords*—**Cellular Automata, Cheating Model, Secret Sharing, Threshold Scheme.

I. INTRODUCTION

SECRET sharing schemes are cryptographic procedures to share a secret among a set of participants in such a way that individual participants can not gain secret, but some qualified subsets of these participants can recover the secret. Secret sharing becomes indispensable whenever secret information needs to be kept collectively by group of participants in such a way that only a qualified subgroup is able to reconstruct the secret. The need for secret sharing arises if the storage system is not reliable, so there is a high likelihood that some pieces of information can be lost [1]. One of the most common secret sharing schemes is threshold scheme. Basic threshold schemes are articulated in [2], [3] where a threshold identifies the number of participants for recreation of the secret. The (k,n) threshold secret sharing scheme divides a secret (s) in to n pieces (shares) and distributes them among the participants, in such a way that any k of them can recover the secret, but any k-1 or fewer can not recover the secret. These schemes are normally set up by a trusted authority (Dealer) who computes shares S_i and assigns it to participants i for a secret $s \in S$ where S is a set of secrets ($Dealer : S \longrightarrow S_1 \times S_2 \times ... \times S_n$). Secret sharing schemes should guarantee that s can only be recovered by a qualified subset of participants (who called active participants) and disqualified participants do not gain any information about the secret. A trusted combiner does the recovery of the secret. The combiner takes an arbitrary collection of shares in order to compute the secret ($Combiner : S_{i_1} \times ... \times S_{i_k} \longrightarrow S$).

Security of secret sharing schemes depends on both combiner and participants' honesty. Unfortunately, the recovered secret can easily be corrupted if dishonest participants (cheaters) send their bogus shares instead of original one. Hence, in the presence of cheaters, the recovered secret is obviously different from the original one. The dishonest participants are able to compute the original secret, while honest participants are not aware of cheating. For instance, Tompa and Woll [4] discovered a way in which dishonest participants can cheat in (k,n) Shamir secret sharing [3]. Therefore, cheating prevention is a desirable characteristic in secret sharing schemes. Some methods of cheating prevention are introduced in [5], [6].

In this paper, we propose a cheating model for the secret sharing schemes based on Linear Memory Cellular Automata (LMCA). Several secret sharing schemes were proposed by utilizing one/two dimensional Cellular Automata (CA) for both text [7] and image [8], [9], [10]. We present a novel uniform model for representation of all these secret sharing schemes. We generalize these schemes using row major numbering method. Each of those schemes can be derived from our generalized scheme. We illustrate proposed cheating model is not contingent upon the neighborhood topology and dimension of CA. We show that collusion of cheaters corrupts the recovered secret. Our cheating model introduces two possible attacks. In the first attack, cheaters send bogus shares and in the second attack they send bogus transition rules of CA. We prove that cheaters are able to compute the cheating value by giving their bogus shares or bogus transition rules

Manuscript received August 31, 2007.

B. Jafarpour is a M.Sc. student in computer engineering from Amirkabir University of Technology, Tehran, Iran (phone: 98-9123895400; e-mail: jafarpour@gmail.com).

A. Nematzadeh is a M.Sc. student in Information Technology engineering from Amirkabir University of Technology, Tehran, Iran (phone: 98-21-4400510; e-mail: azadeh.n@gmail.com).

V. Kazempour is a M.Sc. student in Information Technology engineering from Amirkabir University of Technology, Tehran, Iran (e-mail: kazempour@gmail.com).

B. Sadeghian is an associate professor at Electrical and Computer Engineering Department, Amirkabir University of Technology, Tehran, Iran (e-mail: basadegh@aut.ac.ir.com).

and recover the correct secret.

The rest of this paper is organized as follows. Section 2 demonstrates all secret sharing schemes based on cellular automata. Section 3 outlines our proposed cheating model. The paper is concluded with Section 4 which contains brief recapitulation of the main points and further works.

II. Generalized Secret Sharing Schemes based on Cellular Automata

A cellular automaton [11] consists of a regular array of cells, each in one of finite number of states, which could be represented by *{0, 1, 2, …, c-1}* where c is the number of states. The state of i-th cell at time t is $a_i^{(t)}$. The array can be in any finite number of dimensions. Neighborhood of radius r for every cell in one dimensional CA is $\mathrm{V}_\mathrm{i}^{(t)} = \{a_{i-r}^{(t)},...,a_i^{(t)},...,a_{i+r}^{(t)}\}$ and the Moore neighborhood with radius one for cell *(i,j)* in two dimensional $x \times y$ CA is :

$$V_{i,j}^{(t)} = \{a_{i,j}^{(t)}, a_{i-1,j}^{(t)}, a_{i+1,j}^{(t)}, a_{i,j-1}^{(t)}, a_{i-1,j-1}^{(t)}, a_{i+1,j-1}^{(t)}, a_{i,j+1}^{(t)}, a_{i-1,j+1}^{(t)}, a_{i+1,j+1}^{(t)}\}$$

The state of a cell at time $t+1$ in one dimensional CA is a function of the states of a finite number of cells at time t as follow:

$$a_i^{(t+1)} = f(V_i^{(t)}) \tag{1}$$

The transition function of two-dimensional CA is defined as follow:

$$a_{i,j}^{(t+1)} = f(V_{i,j}^{(t)}), \quad 0 \le \mathrm{i} \le \mathrm{x}-1, \ \ 0 \le \mathrm{j} \le \mathrm{y}-1 \tag{2}$$

The configuration vector of CA at time t is represented by $C^{(t)} = (a_0^{(t)},...,a_{n-1}^{(t)})$. The sequence $\{C^{(t)}\}_{0 \le t \le k}$ is called the evolution of order k of the CA. Suppose ζ is the set of all possible configurations of the CA, the global function of the CA is a transformation which is defined as follow:

$$\phi : \zeta \longrightarrow \zeta, \quad C^{(t+1)} = \phi(C^{(t)}) \tag{3}$$

If ϕ is bijective then there exists inverse cellular automaton with global function ϕ^{-1}. Therefore, backward evolution is possible.

In order to define a generalized notion we use row major numbering method. If row major numbering method is used, all CA transition rules can be considered like formula (1). The only difference is in defining neighborhood. In this numbering method, every cell can be identified by unique index i, where $0 \le \mathrm{i} \le \mathrm{N_{CA}}$. N_{CA} is the number of cells in CA array. For example, Moore neighborhood with radius one in $x \times y$ CA can be rewritten like this:

$$V_i^{(t)} = \{a_i^{(t)}, a_{i-y}^{(t)}, a_{i+y}^{(t)}, a_{i-x}^{(t)}, a_{i-x-y}^{(t)}, a_{i-y+x}^{(t)}, a_{i+x}^{(t)}, a_{i+y-x}^{(t)}, a_{i+y+x}^{(t)}\}$$

$$V_i^{(t)} = \{a_{i+v}^{(t)} \mid v \in I_v\}$$

$$I_v = \{0, -y, +y, -x, -x-y, +x-y, +x, -x+y, +x+y\}$$

I_v is a set of offsets for cell's neighbors in row major numbering.

Therefore, without loss of generality we can use formula (1) for all CA transition rules regardless of dimension and neighborhood. In the reminder of paper, CA represents any n dimensional cellular automata and $n \ge 1$. From now on, we use our generalize notation.

Local transition function for Linear Cellular Automata (LCA) is of the following form:

$$a_i^{(t+1)} = \sum_{v \in I_v} \alpha_v a_{i+v}^{(t)} \pmod{c} \quad 0 \le i \le n-1 \tag{4}$$

α_v is effect of cell's neighbor in computing next state.

According to above transition function, the state of every cell at time $t+1$ depends on the states of its neighbors at time t. In memory cellular automata (MCA) [12], [13], the state of every cell at time $t+1$ is dependent on the states of cells at time t and also some preceded time steps like $t-1, t-2, \dots$. In *k-th* order linear MCA (LMCA), the local transition function is of the following form:

$$a_i^{(t+1)} = \sum_{m=0}^{k-1} f_{m+1}(V_i^{(t-m)}) \pmod{c} \ \ 0 \le \mathrm{i} \le \mathrm{N_{CA}} - 1$$
$$f_{m+1}(V_i^{(t-m)}) = \sum_{v \in I_v} \alpha_v^{k-m-1} a_{i+v}^{t-m} \tag{5}$$

Fd

$f_l, \quad 1 \le l \le k$, is the local transition function of k LCA. Note that the initial configurations of a LMCA is formed by k configurations $C^{(0)},...,C^{(k-1)}$. Evolution starts from these initial configurations. A particular type of reversible LMCA with local transition function (5) is introduced in [7], [8]. They formally proved that the LMCA with formula (5) is reversible if following condition is held:

$$f_k(V_i^{(t-k+1)}) = a_i^{(t-k+1)} \tag{6}$$

And the reverse is:

$$a_i^{(t+1)} = -\sum_{m=0}^{k-2} f_{k-m-1}(V_i^{(t-m)}) + a_i^{(t-k+1)} \pmod{c} \tag{7}$$

Equation (7) can be rewritten as follows:

$$a_i^{(t+1)} = \sum_{m=0}^{k-1} \lambda_m \ f_{k-m-1}(V_i^{(t-m)}) \pmod{c},$$
$$\lambda_m = -1 \ \ if \ \ 0 \le m \le k-2 \tag{8}$$
$$\lambda_m = +1 \ \ if \qquad m = k-1$$

Secret sharing schemes based on CA are (k,n) threshold schemes, in which text or image to be shared, *s*, is one of the initial configurations of a *k-th* order LMCA. In [7] one dimensional CA and in [8], [9], [10] two dimensional CA is used. The rest $k-1$ initial configurations, $C^{(1)},...,C^{(k-1)}$, are random vectors of the same size as *s*. The shares to be distributed among the *n* participants are *n* last consecutive configurations of the evolution of the LMCA. The mentioned schemes [7], [8], [9], [10] are formed by three phases, which are listed below:

1) **The set up phase-** The dealer specifies a random reversible transition function for LMCA that satisfies formula (5) and (6). The secret is considered as the initial configuration ($C^{(0)} = s$). The dealer generates $k-1$ random configurations to complete the initial configurations ($C^{(1)},...,C^{(k-1)}$). Afterwards, the dealer also distributes transition functions to the participants.
2) **The sharing phase-** The dealer chooses an integer number m, such that $m \ge k$ in order to avoid possible overlaps between the initial conditions and the shares. The dealer computes the *(*$m+n-1$*)-th* order evolution of the LMCA: $\{C^{(0)},...,C^{(k-1)},C^{(k)},...,C^{(m)},...,C^{(m+n-1)}\}$. The shares are distributed securely among n participants ($S_0 = C^{(m)},...,S_{n-1} = C^{(m+n-1)}$). Participants can construct the reverse function via formula (7).
3) **The recovery phase-** To recover the secret ($C^{(0)}$), a set of k (of n) consecutive shares of the following form is needed.
$$S_\alpha = C^{(m+\alpha)},...,S_{\alpha+k-1} = C^{(m+\alpha+k-1)}, \quad 0 \le \alpha \le n-k$$
Taking $\widetilde{S}_0 = C^{(m+\alpha+k-1)},...,\widetilde{S}_{k-1} = C^{(m+\alpha)}$, and iterating $m+\alpha+k-1$ times the inverse LMCA, the secret initial configuration, $C^{(0)}$, is obtained.

III. Proposed Cheating Model

In this section we show that the discussed schemes are not secure. Security means correct behavior in face of an intelligent adversary or adversaries [14]. Three important properties in security include confidentiality, integrity, and availability. We show that generalized scheme is not secure in confidentiality and integrity (and therefore schemes in [7], [8], [9], [10] are not secure) since cheater gain correct value of secret and honest participant assumed the corrupted secret as a correct one. We propose a model in which collusion of dishonest participants in the recovery of secret can deceive other participants. In secret recovery phase, honest participants send their valid shares while cheaters send their bogus shares. Cheaters can compute the cheating value contributing in the secret acquired. Cheaters to attain the valid secret can use this value. Honest participant are not aware of cheating and receive corrupted secret assuming it as correct one.

We proposed two different approaches for cheating. The first approach assumes that cheaters fabricate new shares by adding a cheating vector to it. In the second approach, cheaters send bogus transition rules. In both approaches, cheaters convince honest participants of an incorrect secret. In both approaches, we prove that presence of one cheater can corrupt the schemes. If more than one cheater exists, they have to collude in order to obtain the correct secret.

A. Cheating with Bogus Shares

In this attack, participants' shares are represented by set C where C_i is *i-th* participant's share. We suppose that cheaters change their valid assigned shares, C_l to $C_l + \Delta_l \pmod c$ in which $\Delta_l = (\delta_1,...,\delta_n)$ is cheating vector of participant *l*. In fact C_l is original configuration of CA obtained in sharing phase for participant *l* and Δ_l is cheating vector, added to that configuration by cheaters.

Similar to forward function, value of $a_i^{(t+1)}$ in reverse function depends on its neighbors in previous configurations. If some participants cheat and send their bogus shares, $a_i^{(t+1)}$ will have cheating value $\delta_i^{(t+1)}$ that can be shown by equation below:

$$a'^{(t+1)}_i = a_i^{(t+1)} + \delta_i^{(t+1)} \pmod c \tag{9}$$

According to reverse function, cheaters can compute the cheating value for every cell in every step. Theorem 1 shows the cheating value can be computed in first step of reverse function. Computation in other steps to obtain the secret is likewise, after computing preceded configurations.

Theorem 1 – If L is a set of cheaters which is a subset of active participants, and v_i^j is neighbors of cell i in configuration *j*, then cheating value of $a_i^{(t+1)}$ (state of cell i in the configuration $t+1$) can be computed with this formula:

$$\delta_i^{t+1} = [\sum_{l\in L}\sum_{v\in I_v}\alpha_v^{k-l-1}\lambda_{k-l-1}(a'^{\,t-l}_{i+v} - a^{t-l}_{i+v})] \pmod c \tag{10}$$

Proof: First, we assume that only one cheater exists between participants. Computing configurations toward the initial configurations (secret value) is done with the aim of reverse function according to (8):

$$\begin{aligned} a'^{(t+1)}_i = {} & \lambda_k f_k(V_i^{(t-k+1)}) + \lambda_{k-1} f_{k-1}(V_i^{(t)}) + ... + \\ & \lambda_{k-l-1} f_{k-l-1}((V_i^{(t-l)} + \Delta_l) \pmod c) + \\ & ... + \lambda_1 f_1(V_i^{(t-k+2)}) \pmod c \\ & , \quad 0 \le i \le n-1 \; and \; f_k(V_i^{(t-k+1)}) = a_i^{t-k+1} \end{aligned}$$

$$\begin{aligned} a'^{t+1}_i = {} & [\sum_{m=0, m\ne l}^{k-1}[\lambda_{k-m-1} f_{k-m-1}(V_i^{(t-m)})] + \\ & \lambda_{k-l-1} f_{k-l-1}(V_i^{(t-l)} + \Delta_l) \\ & \pm \lambda_{k-l-1} f_{k-l-1}(V_i^{(t-l)})] \pmod c \end{aligned}$$

$$=[\underbrace{\sum_{m=0,m\neq l}^{k-1}[\lambda_{k-m-1}f_{k-m-1}(V_i^{(t-m)})]+\lambda_{k-l-1}f_{k-l-1}(V_i^{(t-l)})}_{a_i^{t+1}}$$

$$+\underbrace{\lambda_{k-l-1}f_{k-l-1}(V_i^{(t-l)}+\Delta_l)-\lambda_{k-l-1}f_{k-l-1}(V_i^{(t-l)})}_{\delta_i^{t+1}}](\bmod c)$$

$$=a_i^{t+1}+\delta_i^{t+1}\ (\bmod c)$$

Above formula shows the cheating value can be computed in the presence of one cheater in this fashion:

$$\delta_i^{t+1}=[\sum_{v\in I_v}\alpha_v^{k-l-1}\lambda_{k-l-1}(a'^{\,t-l}_{i+v}-a^{t-l}_{i+v})](\bmod c)$$

If more than one cheater exists between participants their effect on the cheating value can be easily aggregated:

$$\delta_i^{t+1}=[\sum_{l\in L}\sum_{v\in I_v}\alpha_v^{k-l-1}\lambda_{k-l-1}(a'^{\,t-l}_{i+v}-a^{t-l}_{i+v})](\bmod c)$$

It is noticeable that above formula is obtained regardless of dimension and neighborhood topology of CA. The only condition is that CA use linear transition functions.

With the aim of above theorem, L cheaters ($L\leq k$), contributing in the secret sharing method based on LMCA can obtain the correct secret with computing cheating value while others receive corrupted share without knowing that cheating is done in the background.

B. Cheating with Bogus Transition Rules

In this attack, cheaters send invalid transition rules. Applying invalid transition rules in recovery phase makes preceded configurations toward secret incorrect. We suppose f' as a bogus transition rule that is constructed by adding ω. ω is a cheating vector added to transition rule's parameters. f' can be computed as follow:

$$f'(V_i^t)=\sum_{v\in I_v}(\alpha_v+\omega_v)a_{i+v}^{(t)}\ (\bmod c)$$

$$=\sum_{v\in I_v}\alpha_v a_{i+v}^{(t)}+\sum_{v\in I_v}\omega_v a_{i+v}^{(t)}\ (\bmod c) \qquad (11)$$

$$=f(V_i^t)+\psi(V_i^t)\ (\bmod c)$$

ψ is cheating function that is determined by ω.

During backward evolution, bogus transition rules corrupt configurations. Thus, cheaters should consider both bogus shares and transitions rules while computing the cheating value. Suppose F be the set of invalid configurations and R be the set of bogus transition rules. As a result of applying invalid transition rule, configuration set is partitioned into four separated subsets. Configurations that have:

1. Valid transition rules and valid shares ($F'\cap R'$).
2. Valid transition rules and invalid shares ($F-R$).
3. Bogus transition rules and valid shares ($R-F$).
4. Bogus transition rules and invalid shares ($F\cap R$).

Therefore, the next state of i-th cell should be computed as follow.

$$a_i'^{t+1}=$$

$$\sum_{m\in F'\cap R'}\lambda_{k-m-1}f_{k-m-1}(V_i^{t-m})+\sum_{m\in F-R}\lambda_{k-m-1}f_{k-m-1}(V_i'^{\,t-m})+ \qquad (12)$$

$$\sum_{m\in R-F}\lambda_{k-m-1}f'_{k-m-1}(V_i^{t-m})+\sum_{m\in F\cap R}\lambda_{k-m-1}f'_{k-m-1}(V_i'^{\,t-m})$$

Theorem 2- If L is set of cheaters which is a subset of active participants, who can send bogus transition rules or invalid shares, then cheating value of $a_i^{(t+1)}$ can be computed with this formula:

$$\delta_i^{t+1}=\sum_{m\in F-R}\lambda_{k-m-1}f_{k-m-1}(\Delta_m)+\sum_{m\in R-F}\lambda_{k-m-1}\psi_{k-m-1}(V_i^{t-m})+$$

$$\sum_{m\in F\cap R}\lambda_{k-m-1}f_{k-m-1}(\Delta_m)+\sum_{m\in F\cap R}\lambda_{k-m-1}\psi_{k-m-1}(V_i^{t-m})+ \qquad (13)$$

$$\sum_{m\in F\cap R}\lambda_{k-m-1}\psi_{k-m-1}(\Delta_m)\quad (\bmod c)$$

Proof: Because of linearity of transition functions in (11), it can be concluded easily $f(V_1+V_2)=f(V_1)+f(V_2)$ and $\psi(V_1+V_2)=\psi(V_1)+\psi(V_2)$.

Therefore, equation (12) can be rewritten as follow:

$$a_i'^{t+1}=$$

$$\sum_{m\in F'\cap R'}\lambda_{k-m-1}f_{k-m-1}(V_i^{t-m})+$$

$$\sum_{m\in F-R}\lambda_{k-m-1}f_{k-m-1}(V_i^{t-m})+\sum_{m\in F-R}\lambda_{k-m-1}f_{k-m-1}(\Delta_m)+$$

$$\sum_{m\in R-F}\lambda_{k-m-1}f_{k-m-1}(V_i^{t-m})+\sum_{m\in R-F}\lambda_{k-m-1}\psi_{k-m-1}(V_i^{t-m})+$$

$$\underbrace{\sum_{m\in F\cap R}\lambda_{k-m-1}f_{k-m-1}(V_i^{t-m})}_{a_i^{t+1}}+\underbrace{\sum_{m\in F\cap R}\lambda_{k-m-1}(f_{k-m-1}(\Delta_m)+\psi_{k-m-1}(V_i^{t-m})+\psi_{k-m-1}(\Delta_m))}_{\delta_i^{t+1}}(\bmod c)$$

$$a_i'^{t+1}=a_i^{t+1}+\delta_i^{t+1}\ (\bmod c)$$

Therefore, the cheating value can be computed as follow:

$$\delta_i^{t+1}=\sum_{m\in F-R}\lambda_{k-m-1}f_{k-m-1}(\Delta_m)+\sum_{m\in R-F}\lambda_{k-m-1}\psi_{k-m-1}(V_i^{t-m})+$$

$$\sum_{m\in F\cap R}\lambda_{k-m-1}(f_{k-m-1}(\Delta_m)+\psi_{k-m-1}(V_i^{t-m})+\psi_{k-m-1}(\Delta_m))(\bmod c)$$

Dimension and neighborhood topology of CA do not affect the obtained formula.

L cheaters ($L\leq k$) can compute the cheating value in discussed secret sharing schemes. This value can be used to obtain the valid secret from corrupted one. Honest participants receive corrupted share unaware of cheating.

IV. Conclusion and Future Work

Cellular automata have a very simple structure and can produces very complex emergent behavior with simple transition rules. These features make CA very interesting tool in the field of cryptography. For instance, some secret sharing schemes have been employed recently. In this paper we proposed a new uniform model for representing all secret sharing schemes based on Linear Memory Cellular Automata (LMCA) without considering dimension or neighborhood topology. We used this uniform representation for proposing a cheating model, in which two attacks have been introduced. Salient feature of our cheating model is that it can be applied to all existing secret sharing schemes based on LMCA. Cheaters can diffuse cheating value by corrupting their shares or their local transition rules, along configurations back to shared secret. While honest participants assume that the corrupted secret is valid, cheaters can compute cheating value in shared secret and obtain the valid secret. We showed that how a set of cheaters can work together to compute the cheating value in the initial configurations of cellular automata in both attacks. Methods of preventing and detecting the cheating need to be investigated. Also, other existing cryptological schemes based on CA suspect to be vulnerable to similar attacks. It is worth to revise these cryptological schemes too. We are currently working on designing more secure secret sharing schemes based other models of CA such as heterogeneous CA and nonlinear CA.

REFERENCES

[1] J. Pieprzyk, T. Hardjono, J. Seberry, “Fundamentals of Computer Security,” Springer, 2003.
[2] G. R. Blakley, “Safeguarding cryptographic keys,” in Proc. of 1979 AFIPS National Computer Conference, 1979, pp. 313–317.
[3] A. Shamir, “How to Share a Secret,” Communications of the ACM, Vol. 22. No. 11., 1979, pp. 612 – 613.
[4] M. Tompa, H. Woll, “How to share a secret with cheaters,” Journal of Cryptology, Vol. 1. No. 2. , 1988, pp. 133-138.
[5] H. Ghodosi, J. Pieprzyk, “Cheating prevention in secret sharing,” Australasian Conference on Information Security and Privacy, 2000, pp. 328-341.
[6] J. Pieprzyk, X. Zhang, “Cheating Prevention in Linear Secret Sharing,” Information Security and Privacy. Lecture Notes in Computer Science, Vol. 2384, 2002, pp. 121-131.
[7] A. Rey , J. P. Mateus, G. R. Sanchez, “A secret sharing scheme based on cellular automata,” Applied Mathematics and Computation, Vol. 170, 2005, pp. 1356–1364.
[8] G. A. Maranon, L.H. Encinas, A. M. Rey, “Sharing secret color images using cellular automata with memory,” CoRR 0312034, 2003.
[9] G. Alvarez, A.H. Encinas, L.H. Encinas, A. M. Rey, “A secure scheme to share secret color images,” Computer and Physics communication, Vol. 173. No. 1-2., 2005, pp. 9-16.
[10] G. A. Maranon, L.H. Encinas, A. M. Rey, “A new secret sharing schemes for images based on additive 2-dimensional cellular automata,” Pattern Recognition and Image Analysis. Lecture note in computer science, Vol. 3522, 2005, pp. 411-418.
[11] S. Wolfram, “Cellular Automata, Los Alamos Science,” Vol. 9, 1983, pp. 2-21.
[12] R. Alonso-Sanz, M. Martın, “Elementary cellular automata with memory,” Complex Systems, Vol. 14, 2003, pp. 99–126.
[13] R. Alonso-Sanz, M. Martın, “One-dimensional cellular automata with memory: patterns from a single site seed,” Int. J. Bifur. Chaos Application Science Engineering, Vol. 12, 2002, pp. 205–226.
[14] C. Meadows, “A formal framework and evaluation method for network denial of service,” in Proc. Of 1999 IEEE Computer Security Foundations Workshop, 1998, pp. 4–13.