Reducing Multi-Secret Sharing Problem to Sharing a Single Secret Based on Cellular Automata

Nasrollah Pakniat[1] and Mahnaz Noroozi[2] and Ziba Eslami[2]

[1]Iranian Research Institute for Information Science and Technology (IRANDOC), Tehran, Iran
[2]Department of Computer Science, Shahid Beheshti University, Tehran, Iran

The aim of a secret sharing scheme is to share a secret among a group of participants in such a way that while authorized subsets of participants are able to recover the secret, non-authorized subsets of them obtain no information about it. Multi-secret sharing is the natural generalization of secret sharing for situations in which the simultaneous protection of more than one secret is required. However, there exist some secret sharing schemes for which there are no secure or efficient multi-secret sharing counterparts. In this paper, using cellular automata, an efficient general method is proposed to reduce the problem of sharing k secrets (all assigned with the same access structure and needed to be reconstructed at once) under a certain secret sharing scheme (S), to the problem of sharing one secret under S such that none of the properties of S are violated. Using the proposed approach, any secret sharing scheme can be converted to a multi-secret sharing scheme. We provide examples to show the applicability of the proposed approach.

Keywords- Cryptography; Cellular automata; Secret sharing; Multi-secret sharing; Access structure.

I. Introduction

The concept of *secret sharing* (SS) is independently introduced by Shamir [1] and Blakley [2]. In SS schemes, a secret is divided into some pieces, called secret shadows and then, these shadows are shared among a group of participants. This is done in such a way that any authorized subset of participants can retrieve the secret. A subset is called authorized when it belongs to a predetermined access structure. Many secret sharing applications require the protection of more than one secret. Using an SS scheme multiple times to share each secret separately is the trivial solution to this problem, but in this case each participant should remember too much secret information. In order to reduce the amount of information given to each participant, *multi-secret sharing* (MSS) schemes have been introduced to the literature. Here, we consider the following case which has been introduced by Jackson et al. [3] and many MSS schemes proposed based on it ([4-8]):

A multi-secret sharing (MSS) scheme is a method to share more than one secret among a group of participants in such a way that:

1) *any authorized subset of participants is able to recover all the secrets,*

2) *any non-authorized subset of participants obtains no information about any of the secrets.*

If the two aforementioned conditions are held, but the knowledge on some of the secrets enables the participants in a non-authorized subset to recover some information on other secrets, the scheme would be called a weakly secure MSS scheme. Otherwise, it would be called a strongly secure MSS scheme. In the above definition, a single access structure has been used for all the secrets. The more general case is where each of the secrets is associated with a (possibly) different access structure. The interested readers can find more on the general case in [9], [10].

In recent years, a lot of researches are done to propose SS schemes [4-15]. However, among them, there are some SS schemes for which there are no secure MSS counterpart. For example, the hierarchical threshold SS schemes of [11] are not secure for sharing arbitrary number of secrets. There are also some other SS schemes for which the efficiency of their MSS counterparts are almost as worse as executing them multiple times to share each secret separately. For example, in [12], the authors proposed an MSS scheme in which while the share size of each participant is a constant value, the computational cost and the number of public parameters of this scheme is almost same as when it is used to share each secret separately.

Motivated by these problems, the aim of this paper is to propose a general method to reduce the problem of sharing k secrets (all assigned with the same access structure) under an arbitrary SS scheme (S), to the problem of sharing one secret under S such that all the properties of S (if has any, such as access structure, multi-use, verifiability, etc.) are preserved. It is proved that after applying the proposed method to an SS scheme, the result would be a weakly secure MSS scheme. In this way, any SS scheme can be converted to a weakly secure MSS scheme. The proposed method is based on cellular automata and can be considered as a generalization of the method used in [16, 17], where cellular automata is used to propose secure MSS and secret image sharing schemes for hierarchical threshold access structures [11, 15-19].

It should be noted that it is a while since cellular automata has been used in the context of SS [7, 16, 17, 20-23]. However, a drawback of existing cellular automata-based SS schemes (except for the schemes of [16, 17]) is that their access structure is restricted in the sense that only a set of consecutive participants (based on a predefined ordering) can form an authorized subset. This constraint doesn't apply to the proposed method.

The rest of the paper is organized as follows: In Section II, the basic definitions of memory cellular automata are introduced. The proposed method is described in Section III. In Section IV, the performance and security of the proposed method is analyzed. In Section V, two applications of the proposed method is provided. Finally, the paper is concluded in Section VI.

II. ONE-DIMENSIONAL MEMORY CELLULAR AUTOMATA

A cellular automaton (pl. cellular automata, abbrev. CA) is a discrete dynamical model which consists of a regular grid of cells. The grid can be in any finite number of dimensions and each cell can assume a finite number of states. For each cell, a set of cells called its neighborhood is defined. Then, a local transition function updates the cells simultaneously in discrete time steps. The updated state of each cell is determined via this function in terms of the current state of the cell and the state of the cells in its neighborhood.

The simplest nontrivial CA is a *one-dimensional* CA including an array of N cells with two possible states $s \in \{0,1\}$. For the i-th cell, denoted by $\langle i \rangle$, the symmetric neighborhood of radius r is considered: $\mathcal{N}_i = \{\langle i-r \rangle, \cdots, \langle i \rangle, \cdots \langle i+r \rangle\}$. The *local transition function* of the cellular automata with radius r is of the following form:

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

where $a_i^{(T)}$ denotes the state of $\langle i \rangle$ at time T and $\mathcal{N}_i^{(T-1)} \in \{0,1\}^{2r+1}$ stands for the state of the neighboring cells of $\langle i \rangle$ at time $(T-1)$ (It is assumed that $a_i^{(T)} = a_j^{(T)}$, if $i \equiv j\ (mod\ N)$). A *linear cellular automaton* (LCA) with radius r has a local transition function of the following form:

$$a_i^{(T)} = \sum_{j=-r}^{r} \alpha_j a_{i+j}^{(T-1)}\ (mod\ 2),\ \ 0 \le i \le N-1, \tag{2}$$

where $\alpha_j \in \{0,1\}$ for every j.

Note that there are $2r+1$ neighboring cells for $\langle i \rangle$. Therefore, there exist $2^{2^{2r+1}}$ LCAs where each can be specified by the integer $w = \sum_{j=-r}^{r} \alpha_j 2^{r+j}$ called *rule number*, where $0 \le w \le 2^{2r+1} - 1$.

The vector $C^{(T)} = \left(a_0^{(T)}, \cdots, a_{N-1}^{(T)}\right) \in \{0,1\}^N$ shows the *configuration* of a CA at time T, where $C^{(0)}$ is the *initial configuration*. Moreover, the sequence $\left\{C^{(T)}\right\}_{0 \le T \le k}$ is called the *evolution of order k* of the CA and $\mathcal{C}$ denotes the set of all possible configurations of the CA. The *global function* of the CA is a linear transformation, $\varphi: \mathcal{C} \to \mathcal{C}$, which determines the configuration at the next time step during the evolution of the CA, i.e., $C^{(T)} = \varphi(C^{(T-1)})$.

When there is exactly one past configuration for every current configuration of a cellular automaton, the CA is called *reversible* and the evolution backward is possible. For such CAs, there exists another CA, called its *inverse*, with global function φ^{-1} (see [24]).

The CAs considered so far are memoryless, i.e., the updated state of a cell depends on its neighborhood configuration only at the preceding time step. Nevertheless, one can consider cellular automata for which the state of neighboring cells at time T as well as $T-1, T-2, \cdots$ contribute to determine the state at time $T+1$. This is the concept of the *Memory cellular automaton (**MCA**)* ([25]). Hereafter, by a CA, we mean a particular type of MCA called the k-th order linear MCA ($LMCA$) whose local transition function takes the following form:

$$\begin{aligned} a_i^{(T)} = f_1\left(\mathcal{N}_i^{(T-1)}\right) + f_2\left(\mathcal{N}_i^{(T-2)}\right) + \cdots \\ + f_k\left(\mathcal{N}_i^{(T-k)}\right) (mod\ 2) \end{aligned} \tag{3}$$

where f_j ($1 \le j \le k$) is the local transition function of a particular LCA with radius r . In this case, k initial configurations $C^{(0)}, \cdots, C^{(k-1)}$ are required to start the evolution of $LMCA$.

Furthermore, in order for a memory cellular automaton to be reversible, we have the following proposition proved in [7].

Proposition 1. If $f_k\left(\mathcal{N}_i^{(T-k)}\right) = a_i^{(T-k)}$, then the $LMCA$ given by (3) is reversible and its inverse is another $LMCA$ with the following local transition function:

$$a_i^{(T)} = \sum_{m=0}^{k-2} f_{k-m-1}\left(\mathcal{N}_i^{(T-m-1)}\right) + a_i^{(T-k)}\ (mod\ 2) \tag{4}$$

III. THE PROPOSED SCHEME

In this section, we employ cellular automata to propose a general method to reduce the problem of sharing k secrets (all assigned with the same access structure) to the problem of sharing only one secret. Applying the proposed method to an arbitrary $(M)SS$ scheme (S) would result in an MSS scheme preserving all properties of S (if has any, including access structure, multi-use, verifiability, etc.).

The basic idea behind the proposed approach is to construct an $LMCA$ (M) of order k with the secrets as its initial configurations. Then, M is evolved and a set of k suitably chosen consecutive configurations are chosen, $k-1$ of them would be published and the remaining one would be shared under S. This means that practically, we share only one single secret. In the recovery phase, any qualified subset of participants can use their shares to recover the shared configuration. Then, using the recovered configuration and the public ones, they can reconstruct the inverse cellular automata $\widetilde{M}$ and obtain all k secrets. The general idea is depicted in Fig. 1 with notations as in Table 1.

Let $\{SC_1, \cdots, SC_k\}$ be the set of secrets where each of them is a binary string of length l. Consider S as an arbitrary $(M)SS$ scheme. Now, the MSS scheme obtained by using our reduction and S as its underling $(M)SS$ scheme consists of 3 phases: (1) the setup phase, (2) the sharing phase, and (3) the recovery phase. The details of each phase are provided in the following sections.

A. The setup phase

In this phase, the dealer constructs a reversible $LMCA$ of order k (denoted by M). In details, the dealer performs the following steps:

- Chooses $1 \le r \le \left\lfloor \frac{l-1}{2} \right\rfloor$ as the radius of the symmetric neighborhood of M and publishes it.

- Chooses a random number $1 \leq w_1 \leq 2^{2r+1} - k + 1$ and publishes it. Then, computes the values of $w_i = w_1 + i - 1$ for $1 \leq i \leq k - 1$ as the rule numbers of M.
- Constructs M of order k by

$$a_j^{(T)} = f_{w_1}\left(\mathcal{N}_j^{(T-1)}\right) + \cdots + f_{w_{k-1}}\left(\mathcal{N}_j^{(T-k+1)}\right) + a_j^{(T-k)} \pmod 2 \qquad (5)$$

where $0 \leq i \leq l - 1$ and f_{w_i} is the local transition function of the LCA with radius r and rule number w_i $(1 \leq i \leq k - 1)$. Note that in the proposed scheme each configuration consists of l cells, i.e., the number of cells in each configuration is equal to the size of each secret.

- Sets initial configurations of M as $C^{(0)} = SC_1$, $\cdots$, $C^{(k-1)} = SC_k$.
- Performs the setup phase of S (if applicable).

B. *The sharing phase*

In this phase, the dealer performs the following steps:

- Computes the evolution of M of order $2k - 1$ and obtains $C^{(0)}$,$\cdots$, $C^{(k-1)}$, $C^{(k)}$,$\cdots$, $C^{(2k-1)}$.
- Publishes the values of $\beta_1 = C^{(k)}$,$\cdots$, $\beta_{k-1} = C^{(2k-2)}$.
- Performs the sharing phase of S to share $\beta_k = C^{(2k-1)}$ among participants.

C. *The recovery phase*

On input of the set of shares corresponding to an authorized subset of participants (which belongs to the access structure of S), a trusted party can execute the following steps to recover all the secrets:

- Runs the recovery phase of S to recoverβ_k.
- Constructs $\widetilde{M}$ according to Eq. (4), i.e., the inverse of M, with initial configurations $\tilde{C}^{(0)} = \beta_k$, $\tilde{C}^{(1)} = \beta_{k-1}$, $\cdots$, $\tilde{C}^{(k-1)} = \beta_1$.
- Evolves $\widetilde{M}$ $2k - 1$ times to obtain $SC_1 = C^{(2k-1)}$, $\cdots$, $SC_k = C^{(k)}$.

Table 1. Notations

$SC_1, \cdots, SC_k$	The secrets to be shared.
$P_1, \cdots, P_n$	The participants.
$Sh_1, \cdots, Sh_n$	The shares to be distributed among participants.
r, N	Radius of neighborhood and number of cells in a configuration of the $LMCA$, respectively.

IV. PERFORMANCE AND SECURITY ANALYSIS

In this section, the performance and the security of the MSS schemes obtained by applying the proposed method is analyzed.

A. *Performance analysis*

In this section, the performance of the MSS schemes obtained by using the proposed method is analyzed. The analysis is done in terms of the share size, the computational complexity, the number of public values, and the properties of the obtained MSS schemes.

1) *The share size*

In the MSS schemes obtained by using the proposed method, the share size of each participant is exactly the same as that when S (the underling SS scheme) is used to share only one secret. That is because in the sharing phase of the resulting MSS scheme, the corresponding share to each participant is obtained by sharing only one secret using S.

2) *The computational complexity*

In the proposed method, computing the evolutions of an $LMCA$ is all that is needed to accomplish the desired reduction. It should be noted that in cellular automata-based secret sharing schemes [7, 20-23], it is assumed that the computational complexity of evolving an $LMCA$ is linear in terms of the order of the desired evolution. However, this is not true and it can be easily verified that computing the next configuration of a k-th order $LMCA$ has computational complexity $O(k)$ and therefore, computing the evolution of order k' of this $LMCA$ has computational complexity $O(k(k' - k))$. Using this fact, the achieved reduction has $O(k^2)$ cost where, k is the number of secrets. Therefore, assuming that S is an MSS scheme, based on the computational complexity of S in terms of the number of the secrets, the obtained MSS scheme by using the proposed method can be less or more efficient than S.

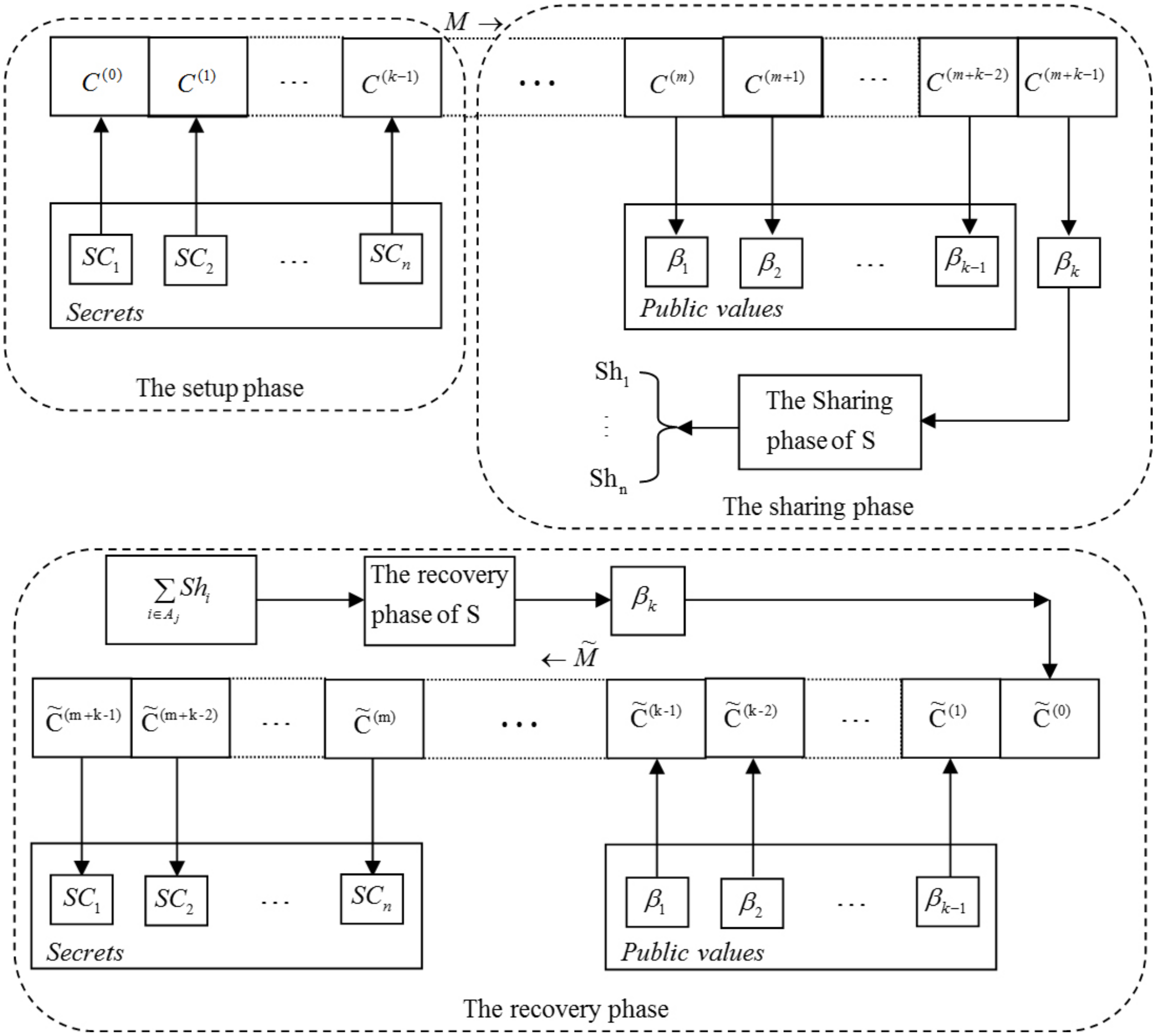

Fig. 1. The diagram of the proposed scheme.

3) *The number of public parameters*

The obtained reduction has the cost of publishing $k-1$ public values. Based on the order of growth of the number of public values of S (assuming that S is an MSS scheme) in terms of the number of the secret, the obtained MSS scheme by applying the proposed method to S can be more or less efficient than the original MSS scheme. For example, in the next section it is shown that applying the proposed method to the SS scheme of [12] would result in a decrease in the number of public parameters, but applying the proposed method to MSS schemes with constant number of public values would result in an increase in the number of public values.

4) *The properties*

Applying the proposed method to S (as the underlying $(M)SS$ scheme) will preserve all the properties of S (if has any, such as multi-use, verifiability, access structure, etc.). The only constraints of the proposed method are that it is not applicable in situations where the secrets are assigned with different access structures or it is required to reconstruct the secrets independently.

B. *The security analysis*

In the following we prove that with knowledge on none of the secrets, non-authorized subsets of participants obtain no information about the secrets. First, we have the following lemma which states a natural property of the memory cellular automata.

Lemma 1. Let M denote a k-th order $LMCA$ and $C^{(j)}, C^{(j-1)}, \cdots, C^{(j-k+1)}$ denote k consecutive configurations of it. Then, without any knowledge about even one of these configurations, it is not possible to obtain any information about $C^{(j+1)}$.

Proof. Without loss of generality assume that there is only one configuration $C^{(j-l)}$ (for a fixed number $0 \leq l \leq k-1$) that we have no information about. In order to compute $C^{(j+1)} = \left(a_0^{(j+1)}, \cdots, a_{N-1}^{(j+1)}\right)$, we have to solve the following system of equations

$$a_i^{(j+1)} = \sum\nolimits_{t=-r}^{r} \alpha_{1,t} a_{i+t}^{(j)} + \cdots + \sum\nolimits_{t=-r}^{r} \alpha_{l+1,t} a_{i+t}^{(j-l)} + \cdots + \sum\nolimits_{t=-r}^{r} \alpha_{k,t} a_{i+t}^{(j-k+1)}, \quad i = 0, \cdots, N-1.$$

In the above system there is $2N$ unknown values ($a_i^{(j+1)}$ and $a_i^{(j-l)}$ for $i = 0, \cdots, N-1$) and only N equations. Therefore, no information about $C^{(j+1)}$ can be obtained. ■

Now, we have the following lemma which is a generalization of Lemma 1 and can be proved easily by induction on i.

Lemma 2. Let M denote a k-th order $LMCA$ and let $C^{(j)}, C^{(j-1)}, \cdots, C^{(j-k+1)}$ denote k consecutive configurations of it. Then, without knowledge about even one of $C^{(l)}$s, $(j-k+1) \leq l \leq j$, it is not possible to obtain any information about any further configurations of M, i.e., it is not possible to obtain information about $C^{(j+i)}$ for any $i \geq 1$.

Now, the following theorem shows that the result of applying the proposed method to an arbitrary secret sharing scheme S (with perfect secrecy) would result in a weakly secure MSS scheme.

Theorem 2. Let S be an arbitrary SS scheme with perfect secrecy. Then, the MSS scheme obtained by applying the proposed method to S is a weakly secure MSS scheme, i.e., in the obtained MSS scheme, with knowledge on none of the secrets, the set of shares corresponding to any non-authorized subset of participants reveals no information about any of the secrets.

Proof. Let $\mathcal{A}$ be an attacker and let B be an arbitrary non-authorized subset of participants. Perfect secrecy of S makes it impossible for $\mathcal{A}$ (even by accessing to the set of shares corresponding to the participants in B) to obtain any information about the shared configuration of M under S in the sharing phase. Therefore, $\mathcal{A}$ has only access to $k-1$ consecutive configurations of M (i.e., the published values). Now, Lemma 2 implies that $\mathcal{A}$ can obtain no information about any further configuration of M . Therefore, he obtains no information about any of the secrets from shares corresponding to B. ■

V. APPLICATIONS FOR THE PROPOSED MEHOD

In this section, the applicability of the proposed method is shown by providing two examples. Since it is straightforward to apply the proposed method to an SS scheme, only the results of these implementations are reported. The interested readers can find the details of the underlying SS schemes in the referred sources.

A. *Applying the proposed method to the SS scheme of [11]*

As explained in the introduction, the MSS scheme presented in [16] is a special case of the proposed method when it is applied to Tassa's hierarchical threshold SS scheme [11]. To the best of our knowledge, it is the only known secure hierarchical threshold MSS scheme. The interested readers can see [16] for the details of this specific implementation.

B. *Applying the proposed method to the MSS scheme of [12]*

In 2010, Das and Adhikari proposed an MSS scheme with general access structure (hereafter, this scheme is denoted for short by DA). DA provide some desirable properties such as verifiability, multi-usability and constant share size. However, the number of public parameters in DA is of order $O(k^2)$ where, k is the number of the secrets. By applying the proposed method to DA as the underlying SS scheme, while the computational complexity of the achieved MSS scheme would be the same as that of the original MSS scheme, the number of public parameters of the scheme would be decreased from $O(k^2)$ to $O(k)$.

VI. CONCLUSIONS

In this paper, cellular automata is employed to propose a general method to reduce the problem of sharing k secrets with a certain secret sharing scheme S, to the problem of sharing one secret under S such that all properties of S remain intact. The proposed method is applicable to situations where the same access structure is assigned to all the secrets and all of them should be recovered at once. Some examples are also provided to show the applicability of the proposed method.

REFERENCES

[1] A. Shamir, How to share a secret, Commun. ACM, 22, 1979, pp. 612-613.

[2] G.R. Blakley, Safeguarding cryptographic keys, AFIPS Conference Proceedings, 48, 1979, pp. 313-317.

[3] W. A. Jackson, K. Martin, and C. ´ OKeefe, Multi-secret Threshold Schemes, Lecture Notes in Comput. Sci., 773, 1994, pp. 126-135.

[4] M. H. Dehkordi and S. Mashhadi, An efficient threshold verifiable multi-secret sharing, Comput. Stand. Inter., 30, 2008, pp. 187-190.

[5] L. J. Pang and Y. M. Wang, A new (t, n) multi-secret sharing scheme based on Shamir's secret sharing, Appl. Math. Comput., 167, 2005, pp. 840-848.

[6] M. H. Dehkordi and S. Mashhadi, New efficient and practical verifiable multi-secret sharing schemes, Inform. Sci., 178, 2008, pp. 2262-2274.

[7] Z. Eslami and J. Z. Ahmadabadi, A verifiable multi-secret sharing scheme based on cellular automata, Inform. Sci., 180, 2010, pp. 2889-2894.

[8] Z. Eslami and S. Kabiri Rad, A New Verifiable Multi-secret Sharing Scheme Based on Bilinear Maps. Wirel. Pers. Commun., 63, 2012, pp. 459-467.

[9] J. Herranz, A. Ruiz, and G. Saez, New results and applications for multi-secret sharing schemes, Designs, Codes and Cryptography, 73, 2013, pp. 1–24.

[10] C. Hu, X. Liao, and X. Cheng, Verifiable multi-secret sharing based on LFSR sequences, Theoretical Computer Science, 45, 2012, pp. 52–62.

[11] T. Tassa, Hierarchical Threshold Secret Sharing, J. Cryptol., 20, 2007, pp. 237-264.

[12] A. Das, A. Adhikari, An efficient multi-use multi-secret sharing scheme based on hash function, Appl. Math. Lett., 23, 2010, pp. 993-996.

[13] S. Mashhadi, M. H. Dehkordi, Two verifiable multi secret sharing schemes based on nonhomogeneous linear recursion and LFSR public-key cryptosystem, Inform. Sci., 294, 2015, pp. 31-40.

[14] C. Hu, , X. Liao, X. Cheng, Verifiable multi-secret sharing based on LFSR sequences, Theor. Comput. Sci., 445, 2012, pp. 52-62.

[15] O. Farras, C. Padro, Ideal hierarchical secret sharing schemes, IEEE T. Inform. Theory, 58, 2012, pp. 3273-3286.

[16] Z.eslami, N. Pakniat, M. Noroozi, Hierarchical Threshold Multi-Secret Sharing Scheme Based on Birkhoff Interpolation and Cellular Automata, The 18th CSI International Symposium on Computer Architecture & Digital Systems (CADS 2015), October 2015.

[17] N. Pakniat, M. Noroozi, Z. Eslami, Secret image sharing scheme with hierarchical threshold access structure, J. Vis. Commun. Image R., 25, 2014, pp. 1093-1101.

[18] N. Pakniat, M. Noroozi, Z. Eslami, Distributed key generation protocol with hierarchical threshold access structure, IET Inform. Secur., 2015, in press.

[19] M. Nojoumian, D. R. Stinson, Sequential Secret Sharing as a New Hierarchical Access Structure, Journal of Internet Services and Info Security (JISIS), Next Gen Networks and Systems Security, 5, 2015 pp. 24-32.

[20] A. M. D. Rey, J. P. Mateus, and G. R. Sanchez, A secret sharing scheme based on cellular automata, Appl. Math. Comput., 170, 2005, pp. 1356-1364.

[21] Z. Eslami, S. H. Razzaghi, J. Z. Ahmadabadi, Secret Image Sharing Based On Cellular Automata and Steganography, Pattern Recognition, 43, 2010, pp. 397-404.

[22] G. Alvarez, L. Hernández Encinas, A. Martín del Rey, A multisecret sharing scheme for color images based on cellular automata, Inform. Sci., 178, 2008, pp. 4382-4395.

[23] X. Wu, D. Ou, Q. Liang, W. Sun, A user-friendly secret image sharing scheme with reversible steganography based on cellular automata, Journal of Systems and Software, 85, 2012, pp. 1852-1863.

[24] T. Toffoli, N. Margolus, Invertible cellular automata: A review, Phys. D, 45, 1990, pp. 229-253.

[25] R. Alonso-Sanz, Reversible cellular automata with memory: two dimensional patterns from a single seed, Phys. D, 175, 2003, pp. 1-30.