Secret image sharing scheme with hierarchical threshold access structure

Nasrollah Pakniat, Mahnaz Noroozi, Ziba Eslami *

Department of Computer Sciences, Shahid Beheshti University, G.C., Tehran, Iran

A R T I C L E   I N F O
Article history:
Received 10 July 2013
Accepted 7 March 2014
Available online 19 March 2014

Keywords:
Cryptography
Secret image sharing
Hierarchical threshold access structure
Cellular automata
Birkhoff interpolation
Information hiding
Reversibility
Tamper detection

A B S T R A C T
A hierarchical threshold secret image sharing (HTSIS) scheme is a method to share a secret image among a set of participants with different levels of authority. Recently, Guo et al. (2012) [22] proposed a HTSIS scheme based on steganography and Birkhoff interpolation. However, their scheme does not provide the required secrecy needed for HTSIS schemes so that some non-authorized subsets of participants are able to recover parts of the secret image. In this paper, we employ cellular automata and Birkhoff interpolation to propose a secure HTSIS scheme. In the new scheme, each authorized subset of participants is able to recover both the secret and cover images losslessly whereas non-authorized subsets obtain no information about the secret image. Moreover, participants are able to detect tampering of the recovered secret image. Experimental results show that the proposed scheme outperforms Guo et al.’s approach in terms of visual quality as well.

© 2014 Elsevier Inc. All rights reserved.

1. Introduction

Sharing images over open channels such as the Internet has attracted considerable attention in recent years [1–8]. However, when it comes to sharing secret images, some challenging problems should be solved first. The first one is the number of parties that can access the secret image. It is definitely a risk to consider a single party due to the accidental or intentional loss/corruption of such images that might occur. On the other hand, if several participants share parts of the secret image, care must be taken to ensure that no malicious shareholder is able to manipulate his/her data. The second concern is the need to keep invaders unaware not only of the content of the secret image itself but also of the very fact that an image is being transferred. Secret sharing schemes which protect and distribute a secret content among a group of participants provide solutions to the first issue. In this regard, the basic example, proposed first by Shamir [9] and Blakley [10], is the concept of a $(t,n)$-threshold secret sharing scheme which encodes a secret data set into $n$ shares and distributes them among $n$ participants in such a way that any $t$ or more of the shares can be collected to recover the secret data, but any $t−1$ or fewer of them provides no information about the secret. Moreover, to ensure recovery of the original secret information some authentication process must be employed so that any manipulation of shares is detected with high probability. To tackle the second concern, steganographic methods are usually employed [11–14]. In these methods, first some innocent looking images, called cover images, are selected. Then the secret data are embedded into cover images and the resulting stego images are distributed among participants using some secret sharing scheme. Clearly, in order not to invoke suspicion, the embedding should create high-quality stego images such that the changes are not visually perceptible. So far, two most popular steganographic methods used in steganographic secret sharing schemes were the least significant bits (LSBs) replacement and the modulus operation.

A method of secret image sharing with steganography and authentication proposed by Lin and Tsai [15] in 2004. Their scheme is an example of a lossy polynomial-based image sharing and the reconstructed secret image may be distorted slightly. Wu et al. [16] in 2004 proposed another scheme in which the secret image is compressed firstly, and then embedded into the cover images by modulus operation. This approach can generate smaller stego images, but the original secret image cannot be retrieved completely in the reconstruction procedure. To recover the secret image losslessly, the method introduced by Thien and Lin [17] can be utilized which splits every pixel with value more than 250 into two pixels. Their method is effective, but the output images are random-looking which attracts the attention of malicious attackers. In order to overcome the defects in Lin and Tsai’s scheme, Yang et al. [18] used Galois field $GF(2^8)$ instead of modulo 251 and
proposed an improved approach in 2007. The scheme proposed by Eslami et al. [4] in 2009 is another example of a secret image sharing scheme with steganography and authentication. Their scheme is an effective lossless sharing scheme based on cellular automata, but their access structure is restricted, i.e., only a subset of some consecutive participants from an ordered set of participants can form an authorized subset. Eslami and Ahmadabadi [5], Ulutas et al. [6] and Yang and Chu [7] are more recent examples on this line of research.

In the above mentioned schemes, it is not possible for participants to recover the cover image losslessly. However, in some applications, such as medical diagnosis, law enforcement, military imaging system, remote sensing and high-energy particle physical experimental investigation, it is important to reverse the stego media back to original after the embedded data is retrieved from it. Therefore, designing a secret image sharing scheme which allows authorized participants to restore the distorted stego image to original without distortion after retrieving the shared data is necessary. Lin and Chan [19] proposed an invertible sharing scheme with steganography to recover the secret image and cover image losslessly. Wu et al. [20] proposed another secret image sharing based on cellular automata and steganography which retrieves the secret and cover images both losslessly.

In reconstruction of the secret image of these schemes, each stego image plays an equivalent role. However, a general threshold access structure can have other useful properties for some applications. For example, when the participants differ in their authority, an access structure which takes this difference into account may be useful. A hierarchical threshold access structure is beneficial in such situations. In a scheme with hierarchical threshold access structure, the secret is shared among a group of participants that is partitioned into levels. The access structure is then determined by a sequence of threshold requirements for these levels, e.g., considering $t_0 < t_1 < t_2 < \ldots$ as the sequence of threshold requirements, a subset of participants is authorized to reconstruct the secret if it has at least $t_0$ participants from the highest level, as well as at least $t_1(> t_0)$ participants from the two highest levels and so forth. In 2007, Tassa proposed a new secret sharing scheme based on Birkhoff interpolation to deal with hierarchical threshold access structures [21]. However, unlike Shamir’s secret sharing scheme, Tassa’s scheme is not able to use all potentials of underlying polynomial to share multiple secrets. Using Tassa’s scheme to share more than $t_0$ secrets makes it possible for some non-authorized subset of participants to recover some of the secrets.

Based on Tassa’s scheme, Guo et al. in [22] proposed a hierarchical threshold secret image sharing scheme with steganographic properties. To the best of our knowledge, their scheme is the only existing hierarchical threshold secret image sharing. In their scheme, after sharing each block of the secret image using Tassa’s scheme, modulus operation is used to hide the shadow data into some cover images. However, their scheme has the following weaknesses:

- As the authors have mentioned in their paper, some non-authorized subsets of participants can obtain parts of the secret image.
- The cover image can not be losslessly recovered.
- There is no authentication in their scheme. Therefore, a malicious participant can make honest participants obtain a fake secret image.
- Compared to existing schemes in the literature with the same threshold parameter, image quality of this scheme is not acceptable (see Section 2.2).

The aim of this paper is to employ cellular automata to propose a hierarchical threshold secret image sharing scheme which overcomes the weaknesses of Guo et al.’s scheme. In the proposed scheme, secret and cover images are recovered losslessly. Moreover, participants are able to check the originality of the recovered secret image. We also formally prove that non-authorized subsets of participants can obtain no information about the secret image. As for the steganographic security, we follow the common methodology considered so far in the context of secret image sharing, i.e., steganographic methods are employed only to prevent noise-like shadow data. Therefore, we consider visual quality of stego images to measure how (visually) susceptible stego images are. The experimental results indicate that the proposed scheme achieves a better visual quality for stego images compared to Guo et al.’s scheme. Despite this, we would like to emphasize that the steganographic method employed in our paper is rather weak (the same as almost all existing literature on steganographic secret image sharing) and well-designed steganalysis algorithms are able to detect the presence of hidden data in our stego images.

The rest of this paper is organized as follows. Section 2 reviews Guo et al.’s hierarchical threshold secret image sharing scheme and discusses its weaknesses. An overview of cellular automata is also provided in this section. In Section 3, we describe the proposed scheme. Security analysis and experimental results of our proposed scheme are provided in Sections 4 and 5, respectively. Finally, the conclusions of this paper are presented in Section 6.

2. Related work

In this section, we first describe Guo et al.’s scheme and then we explain its weaknesses. The necessary background on cellular automata which is the basis of our approach is also covered in this section.

2.1. Review of Guo et al.’s hierarchical threshold secret image sharing scheme

Let $U$ be a group of $n$ participants $P_1, P_2, \ldots, P_n$ divided into $m + 1$ levels $U_0, U_1, \ldots, U_m$ and suppose that the sequence of threshold requirements $t_0, t_1, \ldots, t_m$ determines the hierarchical threshold access structure. Let $SI$ be the secret image and let $C_1, \ldots, C_m$ be the cover images corresponding to $P_1, \ldots, P_m$. The stego image $STG_i$ corresponding to $P_i$ is constructed using $C_i$ and the $P_i$’s share from $SI$, for $i = 1, \ldots, n$. The details of Guo et al.’s scheme are as follows:

**Setup:** The dealer:

1. Chooses a large prime number $p$.
2. Divides $SI$ into $(t_m)$-pixel units $D_1, \ldots, D_l$, where $l = \frac{M_0 \times N_0}{t_m}$ and $M_0$ and $N_0$ are the width and height of the secret image.

**Sharing:** For each unit $D_l (1 \leq l \leq l)$, the dealer:

1. Constructs a $(t_m - 1)$th degree polynomial $F_j(x) = D_j^1 + D_j^2 x + \ldots + D_j^{t_m} x^{t_m-1}(mod\ p)$, where $D_j^i (1 \leq i \leq t_m)$ is $i$th pixel of $D_j$.
2. Assigns to each participant $P_i$ his share from $D_l$ as $SH_j = F_j^{k+1}(i)$, where $k$ is such that $P_l \in U_k$ and $F_j^{k+1}(x)$ is the $(t_m)$th derivative of $F_j(x)$.

**Embedding:** The dealer uses modulus operation to embed each participant’s share from the secret image into his cover image $C_i$ and obtains his stego image $STG_i$.

**Recovery:** Given the stego images corresponding to an authorized subset of participants which satisfy the sequence of threshold requirements, one can recover the secret image as follows:

- Extracts the embedded data from each stego image.
Employs Birkhoff interpolation on extracted data to recover the secret image $SI$.

2.2. Weaknesses of Guo et al.’s scheme

Guo et al. used $t_m$ pixels as coefficients of a polynomial of degree $t_m-1$ in the sharing phase. However, doing so makes it possible for some non-authorized subsets of participants to recover some parts of the secret image. To overcome this problem and as an alternative solution, the authors suggested to share only $t_0$ pixels in each construction of the polynomial. But, this solution increases the amount of the secret shadows and as a result the visual quality of the stego images severely worsens. In their scheme, the cover image cannot be recovered losslessly and hence it’s not useful in applications where this property is needed (see Section 1). Moreover, because of absence of an authentication method in their scheme, a malicious participant can make honest participants obtain a fake secret image and remain unnoticed.

The visual quality of the stego images is related to the amount of shadow data which have to be embedded into the cover images. In most of the secret image sharing schemes in the literature, the amount of shadow data has been related to the threshold parameter, i.e., increasing the threshold parameter makes the amount of shadow data decrease and therefore, the visual quality of stego images will be increased. Therefore, it is not fair to compare visual quality of two schemes, each using different parameters. In [22], the authors compared the result of their scheme using (2, 4, 7) as the sequence of threshold numbers and 10 as the total number of participants, with results of other schemes such as [18] using 3 as threshold number and 4 as the total number of participants. Compared to other schemes, by using the same parameters, the result of implementing Guo et al.’s scheme in ordinary threshold access structures is not acceptable.

2.3. One-dimensional linear memory cellular automata

Guo et al.’s secret image sharing scheme is based on Tassa’s secret sharing scheme. In the sharing phase of Guo et al.’s scheme, the secrets (pixels of the secret image) are the coefficients of some polynomial $F(\cdot)$ (see Section 2). As the authors mentioned in their paper, some unauthorized subsets of participants are able to recover some of the coefficients and therefore, they can obtain some pixels of the secret image. But, the perfect secrecy of Tassa’s scheme makes it impossible for unauthorized subsets of participants to recover the fixed coefficient of polynomial $F(\cdot)$. To secure Guo et al.’s scheme, we need a method that allows a subset of participants to recover pixels of the secret image whenever they are able to recover all of the coefficients of $F(\cdot)$ (including the fixed coefficient).

The method that we use in this paper is based on cellular automata. In the following, we review the definition and properties of cellular automata.

A one-dimensional linear cellular automaton (LCA) is a discrete dynamical model which consists of an array of $N$ cells with two possible states $s \in \{0, 1\}$. For the $i$th cell, denoted by $(i)$, we consider the symmetric neighborhood of radius $r$ which is defined as $N_i = \{i-r, \ldots, i, \ldots, i+r\}$. Then, the state of each cell is updated simultaneously in discrete time steps by means of a local transition function of the following form:

$$a_i(t+1) = \sum_{j \in N_i} z_{ij} a_j(t) \pmod{2}, \quad 0 \leq i \leq N-1.$$  

where $a_i^j$ denotes the state of $(i)$ at time $T$ and $z_{ij} \in \mathbb{Z}$ for every $j$. Furthermore, if $i \equiv j \pmod{N}$, then it is assumed that $a_i^j = a_i^{j'}$ to ensure well-defined dynamics of the CA. Since there are $2r+1$ neighboring cells for $(i)$, there exist $2^{r+1}$ LCAs and each of them can be specified by an integer $w$ called rule number which is defined as follows:

$$w = \sum_{j=-r}^{r} 2^{w_j}.$$  

The configuration of a LCA at time $T$ is shown by the vector $C(T) = (a_{0}^{\ldots}, a_{N-1}^{\ldots})$ where $C(0)$ is the initial configuration. Moreover, the sequence $(C(T))_{T=0}^{\infty}$ is called the evolution of order $k$ of the LCA. The global function of the LCA is a linear transformation, $\Phi$, which determines the configuration at the next time step during the evolution of the LCA, i.e., $C(T+1) = \Phi(C(T))$.

In Memory cellular automaton (MCA) [23] the state of neighboring cells at time $T$ as well as $T-1, T-2, \ldots$ contribute to determine the state at time $T+1$. Hereafter, by a CA, we mean a particular type of MCA called the $r$th order linear MCA (LMCA) whose local transition function takes the following form:

$$a_i(T+1) = \sum_{j=-r}^{r} f_j N_i^{j+r} + \sum_{j=-r}^{r} f_k N_i^{j+r} \pmod{2},$$  

where $f_j$ is the local transition function of a particular LCA with radius $r (1 \leq j \leq t)$ and $N_i^{j+r} \in \{0,1\}$ stands for the state of the neighboring cells of $(i)$ at time $T$. In this case, initial configurations $C(0), \ldots, C(r-1)$ are required to start the evolution of LMCA. A cellular automaton is said to be reversible if for every current configuration of the cellular automaton there is exactly one past configuration. For a reversible CA, there exists another CA, called its inverse, with global function $\Phi^{-1}$. In such CAs the evolution backward is possible (see [24]).

3. The proposed scheme

In this section, a new hierarchical threshold secret image sharing scheme is proposed to overcome the security weakness of Guo et al.’s scheme. We employ cellular automata to achieve this goal. We first provide an overview of the sharing and recovery phase of our approach and then explain each phase in detail.

In order to share a set of $t_m$ secrets, we first set these secrets as initial configurations of a cellular automaton (CA). Then, after required number of evolutions of a properly constructed CA, we get the resulting $t_m$ consequent configurations as temporal secrets and set them as the coefficients of a polynomial. Then, the share of each participant is obtained using appropriate derivative of the polynomial. Now, in order to reconstruct the set of main secrets, first all of the temporal secrets must be reconstructed. This is achieved through the use of Birkhoff interpolation in the scheme. Then, we are able to recover the set of main secrets using the inverse of the CA. The proposed method has the ability to reconstruct the cover image losslessly by sharing the bits of cover images changed during embedding. Moreover, we provide authentication property in the scheme by employing a hash function in the configurations of the CA (instead of bits of cover image) and therefore, we don’t need to embed extra information for authentication purposes.

Suppose that there is a group $U$ of $n$ participants $P_1, P_2, \ldots, P_n$ partitioned into $m+1$ levels $U_0, U_1, \ldots, U_m$ and assume that the sequence of threshold requirements $t_0, t_1, \ldots, t_m$ determines the hierarchical threshold access structure. In the proposed scheme, we have one secret image $SI$ and one cover image $CI$. The stego images $\{STG_{i,j}\}_{0}^{m}$ are produced by embedding the shadow data corresponding to participant $P_i$, into the cover image $CI$. The proposed scheme consists of 4 phases, (1) the setup phase, (2) the sharing phase, (3) the embedding phase and (4) the recovery and authentication phase. In the following, we describe each phase in detail.
3.1. The setup phase

In this phase, the dealer fixes some parameters and constructs a reversible LMCA of order \( t_m \), denoted by \( M \). In what follows, we consider 1 byte for each pixel and we use \( q \) concatenated pixels as a configuration of \( M \). Therefore, the number of cells in each configuration of \( M \) is \( 8 \times q \) and this is why we assume \( 1 \leq r \leq \frac{8 \times q - 1}{q - 1} \). Note also that for a rule number \( w \), we must have \( 0 \leq w \leq 2^{t_m + 1} - 1 \). Here are the detailed steps:

1. Assigns an identity number \( i \) to each participant \( P_i \in U \).
2. Chooses a cryptographic hash function \( H : \{0, 1\}^{8 \times q \times (t_m - 1)} \rightarrow \{0, 1\}^q \).
3. Constructs a reversible LMCA \( (M) \):
   (a) Chooses \( 1 \leq r \leq \frac{8 \times q - 1}{q - 1} \) as the radius of the symmetric neighborhood of the LMCA.
   (b) Chooses a random number \( 0 \leq w_i \leq 2^{t_m + 1} - t_m + 1 \). The rule numbers of the LMCA are then \( w_i, w_i + 1, \ldots, w_i + t_m - 2 \).
   (c) Constructs \( M \) of order \( t_m \) by
      \[
      a_j^{(r+1)} = f_{w_i}(N_j^{(r)}) + \cdots + f_{w_i+t_m-2}(N_j^{(r+2)}) + a_j^{r-1} \pmod{2},
      \]  

where \( 0 \leq j \leq 8 \times q - 1 \) and \( f_{w_i+j} \) is the local transition function of the LMCA with radius \( r \) and rule numbers \( w_i + i, 0 \leq i \leq t_m - 2 \).

3.2. The share generation phase

In the sharing phase, first, the dealer obtains all pixels of the secret image \( SI \), denoted as \( SI = \{s_1, s_2, \ldots, s_{MSI} \} \) and divides \( SI \) into \( (q \times (t_m - 2)) \)-pixel units \( D_1, D_2, \ldots, D_l \) where \( l = \left\lceil \frac{MSI}{q \times (t_m - 2)} \right\rceil \) and \( D_j, 1 \leq j \leq l \) is as follows:

\[
\begin{bmatrix}
D_j^{1,1} & D_j^{1,2} & \ldots & D_j^{1,t_m-2} \\
\vdots & \vdots & \ddots & \vdots \\
D_j^{q,1} & D_j^{q,2} & \ldots & D_j^{q,t_m-2}
\end{bmatrix}
\]

The dealer also divides the cover image \( CI \) into blocks \( D_0^1, D_0^2, \ldots, D_0^l \), where each block contains \( 4 \times q \) pixels. The details of this phase, depicted in Fig. 1(a), are as follows:

The dealer:

1. Divides \( SI \) into \( (q \times (t_m - 2)) \)-pixel units \( D_1, D_2, \ldots, D_l \).
2. Divides \( CI \) into \( (4 \times q) \)-pixel blocks \( D_0^1, D_0^2, \ldots, D_0^l \).
3. Repeats for \( j = 1, \ldots, k \):
   (a) For \( k = 0, \ldots, (t_m - 3) \):

---

**Fig. 1.** Diagram of the proposed scheme. (a) The sharing phase. (b) The recovery phase.
i. Sets initial configuration \(C^{(0)}\) of \(M\) as the result of concatenation of 2 LSBs of pixels in \(j + 1\)th block of the cover image.
2. Computes \(C^{(m-1)}\) as \(H(C^{(0)}, \ldots, C^{(m-2)})\).
3. Computes evolutions of \(M\) of order \(2 \times t_m - 1\) with the initial configurations \(C^{(0)}, \ldots, C^{(m-1)}\) and obtains \(C^{(m)}\).
4. Constructs a (\(t_m - 1\))th degree polynomial \(F_j(x) = C^{(m)} + C^{(m+1)}x + \ldots + C^{(2t_m - 1)}x^{2t_m - 1}\) over \(GF(2^{q/8})\).
5. Assigns to each participant \(P_i\) his share from \(j\)th unit as \(SH_j = F_j^{(s-1)}(i)\), where \(k\) is such that \(P_i \in U_k\).

Remark. As explained in [21], in order to make sure that every authorized subset of participants are able to recover both of the secret and cover images losslessly, we have to choose \(q\) such that:

\[
2^{q/8} > 2^{t_m+2} \times (t_m - 1)^{\frac{t_m}{2}} \times (t_m - 1)! \times n^{\frac{t_m(t_m-1)}{2}}.
\]

### 3.3. The embedding phase

In this phase, the dealer produces final stego images by embedding the data obtained in previous phases into the cover image. To ensure that it is difficult to visually recognize that any data is hidden in the stego images, the embedding procedure must be such that the visual quality of the results have no serious downturn.

We embed the following data in \(CI\) and obtain the stego image \(STG_i\) corresponding to \(P_i\):

- \(i, k_i\): the assigned identity to \(P_i\) and the level to which \(P_i\) belongs.
- \(t = t_0, t_1, \ldots, t_m\): the sequence of threshold numbers.
- \(r, w_i\): the radius of the symmetric neighborhood and the initial rule number of the LMCA.
- \(M_{SI}\) and \(N_{SI}\): the width and height of the secret image.
- \(SH_j\), \(1 \leq j \leq l + 1\): the shares assigned to \(P_i\).

We now outline the details of embedding procedure. The foregoing data, with the same ordering, is considered as an array of elements in \(GF(2^{q/8})\). Each element is embedded into one block of \(CI\) consisting of \(4 \times q\) bytes. Let \((d_1, \ldots, d_{8q})\) be the binary representation of the element which has to be embedded in block \(B\) of \(CI\) with pixels \(X_1, X_2, \ldots, X_{4q}\) with binary representation as in Fig. 2(a). The embedding replaces the least significant bits of \(X_i, X_{2q}, \ldots, X_{4q}\) with \(d_1, \ldots, d_{8q}\) as depicted in Fig. 2(b). Note that the embedding changes at most two of the LSBs in each byte of \(B\). This maintains the quality of stego images.

After obtaining the stego images \((STG_i, i = 1, \ldots, n)\), the dealer sends each stego image to the corresponding participant via a public channel.

Remark. Note that the only aim of using steganography along with the proposed secret image sharing is to prevent distributing noise-like shadow data.

#### 3.4. The recovery and verification phase

The details of this phase are depicted in Fig. 1(b). Suppose that \(t_m\) participants, \(P_{a_1}, \ldots, P_{a_{t_m}}\), pool the stego images \(STG_{a_1}, STG_{a_2}, \ldots, STG_{a_{t_m}}\) to recover the secret image \(SI\). Each \(STG_a\) is divided into a set of blocks with \(4 \times q\) pixels from which the embedded data can be retrieved as follows:

1. For each \(x_i\), \(1 \leq i \leq t_m\):
   - Retrieve from \(STG_a\): \(x_i, kW_i, r\), \(t = (t_0, t_1, \ldots, t_m)\), \(M_{SI}, N_{SI}\), \(SH_j\), \(1 \leq j \leq l + 1\).
2. Check the threshold numbers to verify if these participants are authorized to recover the secret image.
3. Repeat for \(j = 1, \ldots, l\) //reconstruct the \(j\)th unit of the secret image.
(a) Employ Birkhoff interpolation on pairs \((a_i, SH_{a_i})\), \(1 \leq i \leq t_m\), to reconstruct a \(t_m - 1\) degree polynomial \(F_j(x)\). Suppose that \(F_j(x) = a_0 + a_1x + \ldots + a_{t_m-1}x^{t_m-1}\).

(b) Construct the inverse of \(M\), i.e., \(\tilde{M}\), with radius \(r\), rule numbers determined by \(w_i\) and initial configurations:

\[
\tilde{C}_0 = a_{t_m-1}, \quad \tilde{C}_1 = a_{t_m-2}, \ldots, \tilde{C}^{(t_m-1)} = a_0
\]

and evolve \(\tilde{M}\), \(2 \times t_m - 1\) times to obtain \(\tilde{C}^{(t_m)}, \ldots, \tilde{C}^{(2t_m-1)}\).

(c) Check if \(\text{Hash}(\tilde{C}^{(t_m-1)}, \ldots, \tilde{C}^{(2t_m-1)})\) equals \(\tilde{C}_0\) or not.

(d) Divide each \(\tilde{C}^{(i)}(t_m + 2 \leq i \leq 2t_m - 1)\) into \(q\) bytes \(b_i^1, \ldots, b_i^q\).

The pixels of the \(j\)th unit of SI, that is, \(D_j^1, \ldots, D_j^{t_m}, \ldots, D_j^{t_m-2}, \ldots, D_j^{t_m+2}\), are taken as \(b_j^{t_m-1}, \ldots, b_j^{2t_m-1}, \ldots, b_j^{t_m+2}, \ldots, b_j^{2t_m+2}\).

(e) To lossless recovery of the \(j\)th block of the cover image, use binary representation of \(\tilde{C}^{(t_m+1)}\) as depicted in Fig. 2(c).

4. Repeat 3a to 3c on \(\{a_i, SH_{a_i}\}_{1 \leq i \leq t_m}\) and recover the changed bits in the last \(t_m - 1\) changed blocks of the cover image.

5. Restore the changed bits in the last \(t_m - 1\) blocks of the cover image.

4. Security analysis

In this section, we prove that under the assumption of perfect secrecy of Tassa's scheme, the set of stego images corresponding to non-authorized subsets of participants reveals no information about the secret image. We first mention the following theorem which states a natural property of the memory cellular automata. The interested reader can find a proof in [25].

Fig. 3. (a)–(d) The test secret images. (e)–(p) The test cover images.
Theorem 1. Let $M$ denote a $t$th order LMCA. Then, in order to compute $C^{(j+1)}$ for some $j \geq t - 1$, exactly $t$ configurations $C^{(j)}, C^{(j-1)}, \ldots, C^{(j-t+1)}$ are needed.

Now, we prove the following lemma which is a generalization of Theorem 1.

Lemma 1. Let $M$ denote a $t$th order LMCA. Then, without knowing exactly $t$ configurations $C^{(j)}, C^{(j-1)}, \ldots, C^{(j-t+1)}$ for some $j \geq t - 1$, it is not possible to compute any further configurations of $M$, i.e., it is not possible to compute $C^{(j+k)}$ for any $k \geq 1$. 

Fig. 4. An example of the $((2,4,7),10)$ hierarchical threshold case with reversible steganography. (a)-(j) The stego images generated by the proposed scheme. (k) The extracted secret image. (l) The distortion-free recovered cover image.
Proof. We prove by induction on \( k \). Theorem 1 implies that the statement holds for \( k = 1 \). Fix some \( j \) and suppose that the statement holds for some positive integer \( k \). Therefore, it is not possible to compute \( C_{j}^{i(k)} \) without knowing \( t \) configurations \( C_{0}, C_{1}, \ldots, C_{t-1} \). Now, Theorem 1 implies that in order to compute \( C_{j}^{i(k+1)} \) we have to know \( C_{j}^{i(k)}, C_{j}^{i(k-1)}, \ldots, C_{j}^{i(k-t)} \). By the impossibility of computing \( C_{j}^{i(k)} \) from induction hypothesis, we conclude that it is not possible to compute \( C_{j}^{i(k+1)} \) without knowing all of the configurations \( C_{0}, C_{1}, \ldots, C_{t} \). This completes the proof. □

The following theorem shows that the proposed scheme satisfies the security requirement needed in a hierarchical threshold secret image sharing scheme:

**Theorem 2.** Assuming the perfect secrecy of Tassa’s scheme, the proposed scheme is a secure hierarchical threshold secret image sharing scheme, i.e., the set of stego images corresponding to any non-authorized subset of participants reveals no information about the secret image.

**Proof.** Let \( A \) be an attacker and let \( B \) be an arbitrary non-authorized subset of participants. Perfect secrecy of Tassa’s scheme makes it impossible for \( A \) to compute all of the coefficients of \( f(.) \) from the set of stego images corresponding to \( B \). Therefore, \( A \) obtains less than \( m \) consecutive configurations of \( M \). Now, Lemma 1 implies that \( A \) cannot compute any further configuration of \( M \). Therefore, he obtains no information about the blocks of the secret image from stego images corresponding to \( B \). □

5. Experimental results

In this section, we describe some experimental results to demonstrate the characteristics of the proposed scheme. To the best of our knowledge, so far, there has been only one hierarchical threshold secret image sharing scheme in the literature [22]. Therefore, we compare the proposed scheme with this scheme. In order to measure the distortion of the stego images, the peak signal-to-noise ratio (PSNR) can be used:

\[
\text{PSNR} = 10 \times \log_{10} \left( \frac{255^2}{\text{MSE}} \right) \text{dB},
\]

where \( \text{MSE} \) is the mean-square error between the cover image and the stego image. If the cover image is sized \( f \times g \), \( \text{MSE} \) is defined as

\[
\text{MSE} = \frac{1}{f \times g} \sum_{i=1}^{f} \sum_{j=1}^{g} (x_{ij} - y_{ij})^2,
\]

where \( x_{ij} \) and \( y_{ij} \) denote the cover and the stego pixel values, respectively.

In this way, the higher the PSNR values are, the more difficult the visual detection of existence of embedded data in the cover image is.

We perform experiments for \( n = 10 \) and \( m = 2 \), i.e., there are 10 participants divided into 3 levels. Assume that there are 3 participants in the first (highest) level, the second level contains 3 participants and the third (lowest) level includes 4 participants. Assume a sequence of threshold requirements \( t = (t_0, t_1, t_2) = (2, 4, 7) \); that is at least 7 participants have to pool their shares together to reconstruct the secret image (of which at least 4 are from the first two levels and at least 2 are from the first level). In order to demonstrate the visual perception of the stego images, we take “Airplane” with size 256 × 256 as the secret image (Fig. 3(a)) and “Peppers” with size 512 × 512 as the cover image (Fig. 3(p)). Fig. 4(a)–(j) displays the obtained stego images and their PSNR values by using the proposed scheme. The distortion between the cover image and the stego images is slight and therefore, it is difficult for intruders to suspect that some secret data is embedded in the images. If the stego images involved meet the hierarchical threshold access structure, the proposed scheme is able to reconstruct both of the secret and cover images without distortion. The extracted secret and cover images are shown in Fig. 4(k) and (l), respectively.

We performed similar experiments with different test images as cover images. Table 1 displays the PSNR values of the stego images achieved by the proposed scheme “Airplane” with size 256 × 256 (Fig. 3(a)) as the secret image and twelve test images with size 512 × 512 (Fig. 3(e)–(p)) as the cover images. The results show that the PSNR values of the stego images always maintain a steady level and are within [50.87, 52.32].

Table 2 compares the proposed scheme with Guo et al.’s scheme in term of average PSNR values for “Airplane” as the secret image and different cover images. The results show that by using the proposed scheme, we can achieve far better visual quality. That is because, in Guo et al.’s scheme, the authors generated polynomials over GF(p) (where \( p \) is a large prime) and used one pixel as one secret in GF(p). However, in the proposed scheme we generate polynomials over GF(2^{64}) and we use \( q \) concatenated pixels as one secret. Hence, compared to Guo et al.’s scheme, less data must be embedded in our scheme.

Table 3 shows the average PSNR values of the stego images obtained by the proposed scheme using different test secret images (Fig. 3) and different access structures while the cover image is fixed to be Peppers (Fig. 3(p)). The results show that the visual quality of the stego images obtained by the proposed method doesn’t depend on the secret image.

![Table 1](image)

<table>
<thead>
<tr>
<th>Test images</th>
<th>PSNR (dB)</th>
<th>The first level</th>
<th>The second level</th>
<th>The third level</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
<td>1</td>
<td>2</td>
<td>3</td>
</tr>
<tr>
<td>Baboon</td>
<td>51.24</td>
<td>51.22</td>
<td>51.26</td>
<td>51.20</td>
</tr>
<tr>
<td>Barbara</td>
<td>51.17</td>
<td>51.15</td>
<td>51.11</td>
<td>51.09</td>
</tr>
<tr>
<td>Boat</td>
<td>51.00</td>
<td>51.04</td>
<td>50.98</td>
<td>51.00</td>
</tr>
<tr>
<td>Cameraman</td>
<td>51.90</td>
<td>51.93</td>
<td>51.94</td>
<td>51.94</td>
</tr>
<tr>
<td>Couple</td>
<td>50.93</td>
<td>50.94</td>
<td>50.92</td>
<td>50.99</td>
</tr>
<tr>
<td>Elaine</td>
<td>50.88</td>
<td>50.90</td>
<td>50.88</td>
<td>50.90</td>
</tr>
<tr>
<td>Girl</td>
<td>50.93</td>
<td>50.88</td>
<td>50.88</td>
<td>50.89</td>
</tr>
<tr>
<td>House</td>
<td>52.32</td>
<td>52.30</td>
<td>52.31</td>
<td>52.27</td>
</tr>
<tr>
<td>Lake</td>
<td>51.12</td>
<td>51.10</td>
<td>51.11</td>
<td>51.14</td>
</tr>
<tr>
<td>Lena</td>
<td>51.12</td>
<td>51.11</td>
<td>51.08</td>
<td>51.11</td>
</tr>
<tr>
<td>Man</td>
<td>51.11</td>
<td>51.10</td>
<td>51.11</td>
<td>51.15</td>
</tr>
<tr>
<td>Peppers</td>
<td>51.03</td>
<td>51.09</td>
<td>51.05</td>
<td>51.07</td>
</tr>
</tbody>
</table>
The average analysis algorithms. The proposed scheme has the weaknesses of Guo et al.'s scheme. We also employ steganography we propose a secret image sharing scheme which overcomes

Table 2
Comparisons of optimal image quality between the proposed scheme and Guo et al.'s scheme for different cover images.

<table>
<thead>
<tr>
<th>Schemes</th>
<th>Average PSNRs</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>Baboon</td>
</tr>
<tr>
<td>Ours</td>
<td>51.23</td>
</tr>
<tr>
<td>Guo et al.'s</td>
<td>38.19</td>
</tr>
</tbody>
</table>

$\left\{\begin{array}{ll}
\text{n} & : \text{The total number of participants.} \\
\text{M} & : \text{the number of hierarchical levels.} \\
\text{NPL} & : \text{the number of participants in each of M hierarchy levels.} \\
\text{t} & : \text{The required sequence of threshold numbers to reconstruct the secret image.}
\end{array}\right.$

6. Conclusion
In this paper, by using cellular automata and Birkhoff interpolation we propose a secret image sharing scheme which overcomes the weaknesses of Guo et al.'s scheme. We also employ steganography to prevent noise-like shares. The proposed scheme has the following advantages:

- It admits a hierarchical threshold access structure.
- It is able to recover both of the secret and cover image losslessly.
- After lossless recovery of the secret and cover image, participants are able to check the validity of the secret image, i.e., they are able to detect whether stego images are tampered or not.
- The set of stego images corresponding to a non-authorized subset of participants reveals no information about the secret image.
- Compared to Guo et al.'s scheme, the stegos produced by the proposed scheme have better visual quality.

However, the same as almost all existing steganographic secret image sharing schemes, our method is not secure against steganalysis algorithms.

References