Hindawi Publishing Corporation
Mathematical Problems in Engineering
Volume 2015, Article ID 849768, 11 pages
http://dx.doi.org/10.1155/2015/849768

Research Article

A Reversible Steganography Scheme of Secret Image Sharing Based on Cellular Automata and Least Significant Bits Construction

Wei-Tong Hu, Ming-Chu Li, Cheng Guo, and Li-Feng Yuan

School of Software, Dalian University of Technology, No. 8 Road, Jinzhou District, Dalian 116620, China

Correspondence should be addressed to Cheng Guo; guo8016@gmail.com

Received 11 November 2014; Accepted 7 April 2015

Academic Editor: Ivan D. Rukhlenko

Secret image sharing schemes have been extensively studied by far. However, there are just a few schemes that can restore both the secret image and the cover image losslessly. These schemes have one or more defects in the following aspects: (1) high computation cost; (2) overflow issue existing when modulus operation is used to restore the cover image and the secret image; (3) part of the cover image being severely modified and the stego images having worse visual quality. In this paper, we combine the methods of least significant bits construction (LSBC) and dynamic embedding with one-dimensional cellular automata to propose a new lossless scheme which solves the above issues and can resist differential attack and support parallel computing. Experimental results also show that this scheme has the merit of big embedding capacity.

1. Introduction

Secret sharing scheme is devoted to protecting secret information from being lost, destroyed by attackers, stolen by illegal users, and so on. It was independently proposed by Shamir [1] and Blakley [2] in 1979. The former one is based on the polynomial interpolation, and the latter one is based on the intersections of some high dimensional planes in a high dimensional space.

In the early research period of secret sharing scheme, researchers focused on designing different types of secret sharing schemes for small secret data. Later, secret image sharing, which combines cryptography and image processing techniques, becomes an active research area. Suppose that a secret image sharing scheme has (k, n)-threshold, where $k \leq n$, in which a secret image is divided into n shadows distributed to n participants. The secret image can only be restored by k or more shadows; but no one can reveal any information about the secret image with any $k-1$ or fewer shadows.

Visual cryptography scheme (VCS) is an important category of secret image sharing scheme, which was introduced by Naor and Shamir [3] in 1995. VCS has the merit of stacking-to-see property but with the disadvantages of the large pixel expansion and low visual quality of the restored image. In VCS, any k participants could photocopy their shadows on transparencies and stack them on an overhead projector to decode the secret visually through human visual system without the aid of any hardware or computation. However, no one can reveal any information about the original secret with any $k-1$ or fewer shadows. Based on Droste [4] work, many researchers have made lots of efforts to do further research on VCS from multiple aspects. Some schemes focused on reducing the pixel expansion [4–7], some on improving the visual quality of the restored image [8–11], some on sharing gray and chromatic secret images [12–14], and so on. VCS's merit is apparent from some aspects or in some application areas. But it still has the inherent defects such as pixel expansion, poor visual quality of the restored image, and meaningless shadows.

The secret image sharing scheme with Lagrange's interpolation and steganography can solve the above three inherent defects of VCS, which is an important category of secret image sharing scheme. Until now, two most popular steganographic methods are least significant bits replacement (LSBR) and modulus operation. The shared noise-like secret data is concealed into the cover image with meaningful content. These modified cover images are called stego images. In 2004, Lin and Tsai [15] proposed a polynomial-based secret image sharing scheme with steganography and authentication. This scheme is lossy because it uses LSBR method to embed secret data into the cover image. Hence, the cover image cannot be recovered losslessly. And the restored secret image by this scheme may be distorted slightly. The reason for the lossy restoration of the secret image is that such kind of schemes truncate the pixel values of the secret images into the interval $[0, 250]$. In addition, quantization-based schemes, such as [16, 17], just gain the lossy secret image due to the quantization errors.

To restore the secret image losslessly, the schemes of [18–20] used two pixels to replace any pixel whose value is greater than or equal to 250. Obviously, this will lead to pixel expansion and increase the amount of secret data which will be embedded into the cover image. In 2007, Yang et al. [21] utilized Galois Field $GF(2^8)$ instead of using modulo 251 operation to avoid the way of dividing a pixel whose value is greater than or equal to 250 into two pixels. Hence, the secret image could still be restored losslessly without pixel expansion. Hu et al. [22] and Li et al. [23] also used Galois Field $GF(2^8)$ to guarantee the secret image to be restored completely.

Lossless recovery of the secret image is an important measure in secret image sharing field. Although many schemes in this field have achieved the lossless recovery of the secret image, there are just a few schemes to restore both the secret image and the cover image losslessly, such as [22, 24–27]. The schemes of [24, 26, 27] use modulus operation to embed secret data into the cover image, and the scheme of [25] utilizes extra Sudoku table to conceal secret data into the cover image. However, the scheme of [22] uses the new LSBC method to do the embedding work. Modulus operation often leads to overflow or underflow issues. The schemes of [24, 26, 27] need to do extra work to avoid the situation that the secret image and the cover image cannot be recovered completely under some circumstances; the scheme of [26] converts all secret pixels into the values in m-ary notational system and requires that any pixel whose value is within $[\lfloor 255/m \rfloor \times m, 255]$ must be skipped in order to recover the secret image and the cover image completely, where $3 \leq m < 255$ and m is a prime number. Therefore, modulus operation has such kind of limitations. Additional storage is needed by the method of using Sudoku table. Another defect in the scheme of [25] is that it uses the two fixed least significant bits in each pixel from the cover image to embed secret data. It does not have the flexibility like modulus operation by which one or more least significant bits could be used to embed secret data with different modulus values. The scheme of [22] does not require extra work to adapt to some pixels with special values and needs no additional storage for Sudoku table or others. Meanwhile, it also possesses the flexibility as modulus operation. All the schemes of [22, 24–26] use polynomial evaluation and interpolation algorithm with computation complexity as high as $O(n\log^2 n)$ [28]. However, only the scheme of [27] and our scheme utilize cellular automata whose computation complexity is only $O(n)$.

Additionally, in secret image sharing field, the visual quality of stego images is also an important measure. The schemes of [24–27] embed secret data into the cover image sequentially and sometimes partial areas of the cover image are modified severely while the other areas of it are untouched. Thus the visual quality of these generated stego images will be poor with the above traditional embedding method. The scheme of [22] and our scheme use the method of dynamic embedding. This method will calculate the size of the cover image and the amount of the secret data to adjust the way of LSBC and embed secret data smoothly. Then the visual quality of the stego images can be improved.

The remaining part of this paper is organized as follows. Section 2 reviews the basic definitions of one-dimensional cellular automata. The proposed scheme is introduced in Section 3, and experimental results and comparisons are given in Section 4. Section 5 concludes this paper.

2. One-Dimensional Memory Cellular Automata

One-dimensional finite boolean cellular automata (CA for short) are discrete dynamical systems constructed by a finite array of N identical objects called cells [29]. The state value s of each cell is 0 or 1. The evolution of CA is that the states of all cells are updated synchronously in discrete time steps according to a local transition function. The update of each cell's state relies on variables of this function. These variables are the previous states of the cell itself and its neighbor cells. Let $\langle i \rangle$ denote the ith cell and $a_i^{(T)}$ the state of $\langle i \rangle$ at time step T. Symmetric neighborhood of radius r is defined as $N_i = \{\langle i - r\rangle, \dots, \langle i \rangle, \dots, \langle i + r\rangle\}$. Thus, the local transition function of the cellular automata with radius r is represented as follows:

$$\begin{aligned} a_i^{(T+1)} &= f\left(N_i^{(T)}\right) \\ &= f\left(a_{i-r}^{(T)}, \dots, a_i^{(T)}, \dots, a_{i+r}^{(T)}\right), \quad 0 \leq i \leq N-1, \end{aligned} \tag{1}$$

where $N_i^{(T)} \subset (Z_2)^{2r+1}$ stands for the states of the neighbor cells of $\langle i \rangle$ at time T. Additionally, periodic boundary condition is used in this paper; namely, if $i \equiv j (\operatorname{mod} N)$, then $a_i^{(T)} = a_j^{(T)}$.

The configuration of one-dimensional CA at time step T is defined as the vector $C^{(T)} = (a_0^{(T)}, \dots, a_{N-1}^{(T)})$, while $C^{(0)}$ denotes the initial configuration. Furthermore, the sequence $\{C^{(T)}\}_{0 \leq T \leq k}$ is called the evolution of the kth CA, and let Γ represent the set of all possible configurations of the CA.

The global function of the CA is a linear transformation; namely, $\Phi : \Gamma \rightarrow \Gamma$, which generates the configuration at the next time step in the evolution of the CA; that is, $C^{(T+1)} = \Phi(C^{(T)})$. If Φ is bijective, there exists the current CA's inverse whose global function is Φ^{-1}, and the current CA is called reversible. In such CAs, the reverse evolution is possible [30].

The local transition function of linear cellular automata (LCA) with radius r is defined as follows:

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

where $\lambda_j \in Z_2$ for each j. Since $\langle i \rangle$ has $2r + 1$ neighbor cells, there exist $2^{2^{r+1}}$ LCAs. Each LCA can be specified by an integer ω called rule number which is defined as follows:

$$\omega = \sum_{j=-r}^{r} \lambda_j 2^{r+j}, \tag{3}$$

where ω is in the interval $[0, 2^{2^{r+1}} - 1]$. Then f_ω is defined to represent the local transition function of the CA with rule number ω.

The CA considered above is memoryless. For example, the updated state of a cell at time step $T + 1$ only depends on its neighbor configuration at time step T. However, one can consider cellular automata for which the state of each cell at time $T+1$ depends on the states of its neighbor cells at time T as well as $T-1, T-2, \ldots$. This kind of CA is called memory cellular automata (MCA). In fact, there also exists a special type of MCA called the kth order linear MCA (LMCA) whose local transition function is expressed as follows:

$$\begin{aligned} a_i^{(T+1)} &= F\left(N_i^{(T)}, \ldots, N_i^{T-k+1}\right) \\ &= f_{\omega_1}\left(N_i^{T}\right) + \cdots + f_{\omega_k}\left(N_i^{T-k+1}\right) (\operatorname{mod} 2), \end{aligned} \tag{4}$$

where $0 \le i \le N - 1$. Since the arithmetic operation is performed in Z_2, (4) takes the way of modulo 2. k initial configurations $C^{(0)}, \ldots, C^{(k-1)}$ are needed to evolve the kth order LMCA.

If $f_{\omega_k}(N_i^{(T-k+1)})$ equals a_i^{T-k+1} in (4), then it is easy to construct a reversible LMCA. Here, the local transition function of the kth order reversible LMCA is defined as follows:

$$\begin{aligned} a_i^{T+1} &= f_{\omega_1}\left(N_i^{(T)}\right) + \cdots + f_{\omega_{k-1}}\left(N_i^{(T-k+2)}\right) \\ &\quad + a_i^{(T-k+1)} (\operatorname{mod} 2). \end{aligned} \tag{5}$$

And its inverse CA is another kth order LMCA with the following local transition function:

$$\begin{aligned} a_i^{T+1} &= f_{\omega_{k-1}}\left(N_i^{(T)}\right) + \cdots + f_{\omega_1}\left(N_i^{(T-k+2)}\right) \\ &\quad + a_i^{(T-k+1)} (\operatorname{mod} 2). \end{aligned} \tag{6}$$

For the proof of this kind of reversible LMCA, please refer to [31].

3. The Proposed Scheme

In this part, we will elaborate this new reversible steganography scheme of secret image sharing, which is based on cellular automata and LSBC. LSBC is a method to construct a new digit number by a few bits from different pixels, which will be used to join polynomial computations, evolutions of cellular automata, or other computations. Figure 1 shows an example of the way to construct a new digital number. This scheme includes three phases: (1) setup phase: the dealer determines the initial configurations, constructs the reversible LMCA, and determines the related parameters about dynamic embedding; (2) sharing and embedding phase: the dealer evolves the LMCA to generate shared bits and embeds them into the cover image to form the stego images; (3) recovering phase: shared bits are retrieved from the stego images and used as the initial configurations of the reverse LMCA. Then the secret image and the cover image are restored by evolving the reverse LMCA. Moreover, it is supposed that the bit depth of the involved cover images and secret image is 8 in this paper.

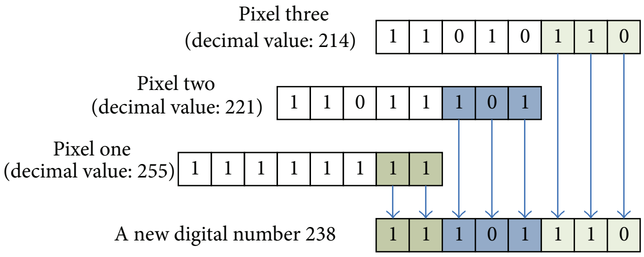


Figure 1: Example of the way to construct a new digital number by LSBC.

3.1. Setup Phase. In this phase, the dealer constructs the kth order reversible LMCA according to the following steps.

(1) Since the bit depth of each secret pixel is 8, the dealer determines that the radius r of the symmetric neighborhood of the LMCA is in the interval [1, 3], and the cell sum N in each configuration is 8.

(2) Select and publish random numbers $\omega_1, \ldots, \omega_{k-1}$, where $1 \le \omega_j \le 2^{2r+1} - 1$ for $1 \le j \le k - 1$. These numbers are produced by secure cryptographic pseudorandom number generators. For details about it, please refer to [32].

(3) The constructed reversible LMCA is as follows:

$$\begin{aligned} a_i^{T+1} &= f_{\omega_1}\left(N_i^{(T)}\right) + \cdots + f_{\omega_{k-1}}\left(N_i^{(T-k+2)}\right) \\ &\quad + a_i^{(T-k+1)} (\operatorname{mod} 2), \end{aligned} \tag{7}$$

where $0 \le i \le N - 1$, and f_{ω_j} is the local transition function of the LMCA with rule number ω_j, $1 \le j \le k - 1$.

Additionally, the dealer will determine the values of CIPNL and CIPNH according to the size values of the cover image and the secret image, and the threshold value k. For details about the calculation of CIPNL and CIPNH, please refer to [22]. We define the total times of using CIPNL pixels and that of CIPNH pixels to construct s_0 as x and y, respectively. If y is bigger, more pixels will be used to construct s_0 each time and every pixel of this block will suffer from less modification. Then the visual quality of stego images can improve. For details about the calculation of x and y, please refer to [22].

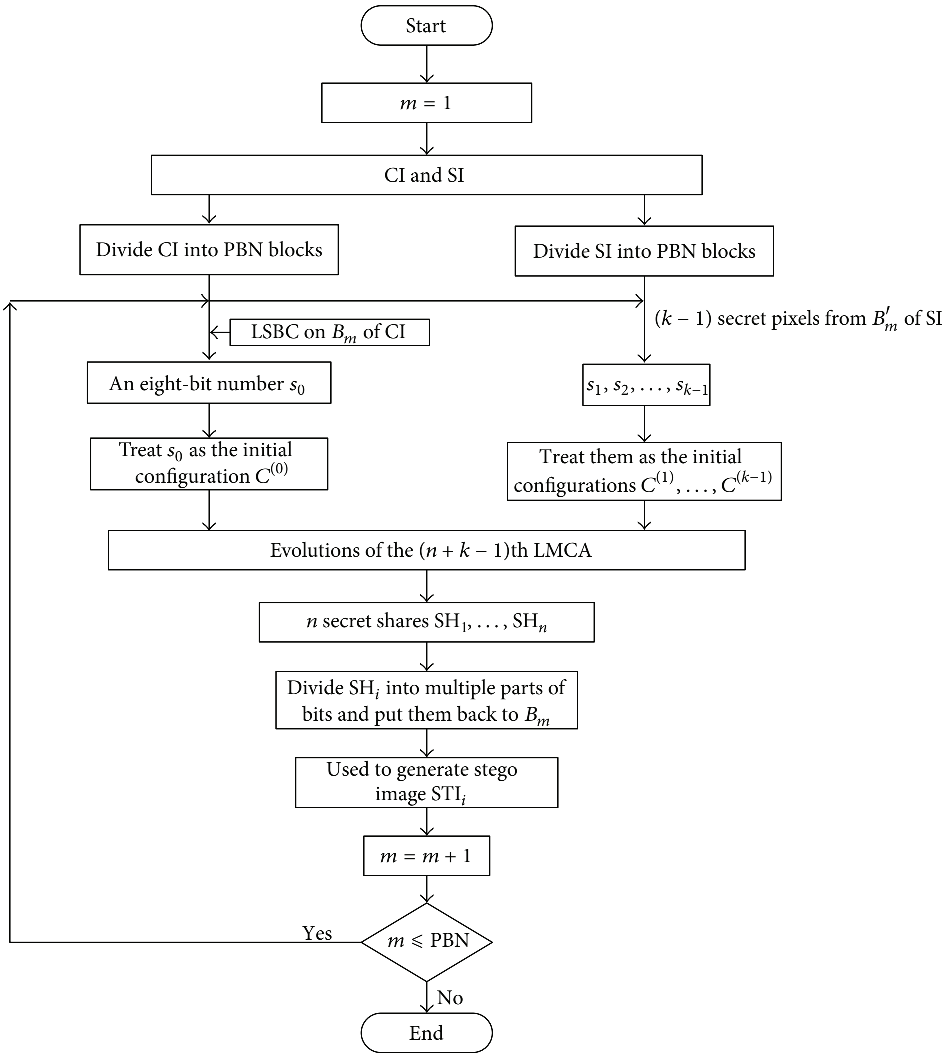

Start
$m = 1$
CI and SI
Divide CI into PBN blocks
Divide SI into PBN blocks
LSBC on B_m of CI
$(k-1)$ secret pixels from B'_m of SI
An eight-bit number s_0
$s_1, s_2, \dots, s_{k-1}$
Treat s_0 as the initial configuration $C^{(0)}$
Treat them as the initial configurations $C^{(1)}, \dots, C^{(k-1)}$
Evolutions of the $(n+k-1)$th LMCA
n secret shares $SH_1, \dots, SH_n$
Divide SH_i into multiple parts of bits and put them back to B_m
Used to generate stego image STI_i
$m = m + 1$
Yes
$m \leqslant PBN$
No
End

Figure 2: Flow chart of the sharing and embedding phase.

3.2. Sharing and Embedding Phase. Firstly, the dealer calculates the number of the secret pixel blocks. Let us define it as PBN, which has the following form:

$$\mathrm{PBN} = \left\lceil \frac{\mathrm{SIW} \times \mathrm{SIH}}{(k-1)} \right\rceil . \tag{8}$$

Each block contains $k-1$ secret pixels. If the last block does not have $k-1$ pixels, it will be padded with 0. Secret data derived from a secret pixel block will be embedded into a pixel block from the cover image. Hence, the dealer needs to retrieve PBN pixel blocks from the cover image, with each block containing CIPNL or CIPNH pixels. The pixel blocks from the cover image could be defined as $B_1, \dots, B_m$, $1 \le m \le \mathrm{PBN}$. For each B_m, LSBC method will be used to extract least significant bits from pixels of it to construct s_0. This s_0 will be used as the initial configuration $C^{(0)}$. Other $k-1$ configurations of the LMCA are filled by the $k-1$ secret pixels. Then the evolutions of the LMCA will generate n shares $SH_1, \dots, SH_n$ corresponding to n participants. Each share has the same size as s_0. Finally, the dealer will embed the secret shares back to the pixel blocks of the cover image. After processing all the pixels of the secret image, n stego images will be generated and the dealer will distribute one stego image to each participant. Figure 2 shows the flow chart of the sharing and embedding phase. Detailed description of this process is formulated as follows.

(1) Divide the secret image into PBN blocks with each block containing $k-1$ pixels.

(2) Repeat Step (2.1) to Step (2.4) for $m = 1, \dots, \mathrm{PBN}$.

(2.1) Use LSBC to retrieve least significant bits from B_m to construct s_0, and use s_0 as the initial configuration $C^{(0)}$ of the LMCA.

(2.2) Take the $k-1$ pixels of the mth secret pixel block as the initial configurations $C^{(1)}, \dots, C^{(k-1)}$ of the LMCA.

(2.3) With the initial configurations $C^{(0)}, \dots, C^{(k-1)}$, the evolutions of the $(n+k-1)$th LMCA are

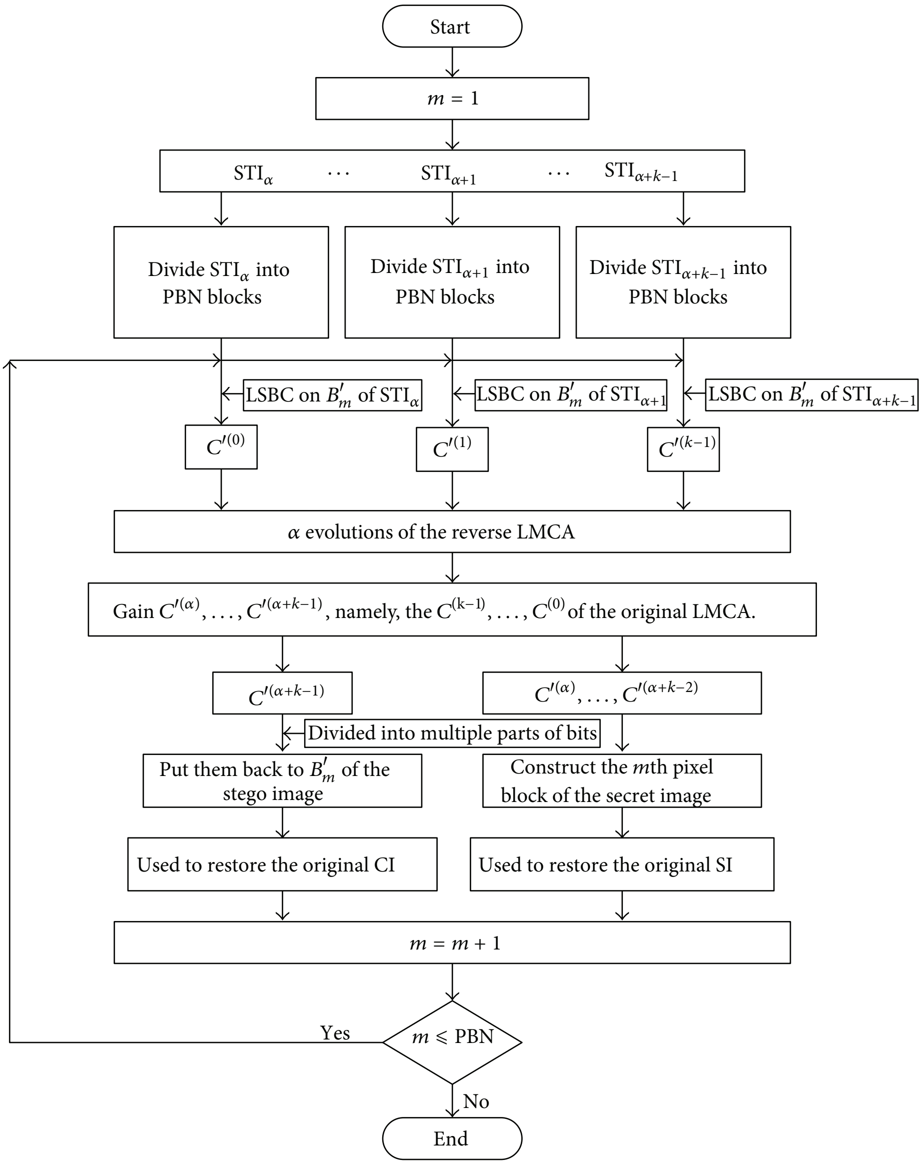

Start
$m = 1$
STI_{α} ⋯ $\mathrm{STI}_{\alpha+1}$ ⋯ $\mathrm{STI}_{\alpha+k-1}$
Divide STI_{α} into PBN blocks
Divide $\mathrm{STI}_{\alpha+1}$ into PBN blocks
Divide $\mathrm{STI}_{\alpha+k-1}$ into PBN blocks
LSBC on B'_m of STI_{α}
LSBC on B'_m of $\mathrm{STI}_{\alpha+1}$
LSBC on B'_m of $\mathrm{STI}_{\alpha+k-1}$
$C'^{(0)}$
$C'^{(1)}$
$C'^{(k-1)}$
α evolutions of the reverse LMCA
Gain $C'^{(\alpha)}, \ldots, C'^{(\alpha+k-1)}$, namely, the $C^{(k-1)}, \ldots, C^{(0)}$ of the original LMCA.
$C'^{(\alpha+k-1)}$
$C'^{(\alpha)}, \ldots, C'^{(\alpha+k-2)}$
Divided into multiple parts of bits
Put them back to B'_m of the stego image
Construct the mth pixel block of the secret image
Used to restore the original CI
Used to restore the original SI
$m = m + 1$
Yes
$m \leqslant \mathrm{PBN}$
No
End

Figure 3: Flow chart of the recovering phase.

calculated to generate $C^{(k)}, \ldots, C^{(n+k-1)}$, which correspond to n secret shares $\mathrm{SH}_1, \ldots, \mathrm{SH}_n$.

(2.4) Divide SH_i into multiple parts of bits and put them back to B_m. In other words, all the bit positions in B_m, where the bits to construct s_0 come from, are filled with new bits in this step. This modified block will be used to construct STI_i.

(3) If some pixels in CI are not modified, namely, that the embedding capacity of CI has not been fully utilized, then assign them to n stego images STI_is directly.

(4) The dealer distributes the meaningful stego image STI_i to the participant P_i, $1 \le i \le n$.

In Step (2.3), for simplicity, this scheme evolves n times on the basis of k initial configurations. Obviously, more evolution steps can be added into the evolution of the LMCA.

3.3. Recovering Phase. Cellular automata scheme requires that at least k stego images are sequential when recovering the secret image and the cover image. Figure 3 shows the flow chart of the recovering phase. Suppose that k stego images $\mathrm{STI}_{\alpha}, \mathrm{STI}_{\alpha+1}, \ldots, \mathrm{STI}_{\alpha+k-1}$ are collected ($1 \le \alpha \le n - k + 1$); the steps to recover the secret image and the cover image are as follows.

(1) Get PBN pixel blocks from every stego image with each block containing CIPNL or CIPNH pixels. Define these blocks as $B'_1, \ldots, B'_m$, $1 \le m \le \mathrm{PBN}$.

(2) Repeat Step (2.1) to Step (2.3) for $m = 1, \ldots, \mathrm{PBN}$.

Table 1: The image quality of the stego images for various cover images with $k = 3$.

Cover images	Stego image 1		Stego image 2		Stego image 3	
	PSNR (dB)	SSIM	PSNR (dB)	SSIM	PSNR (dB)	SSIM
Airplane	51.138	0.9955	51.144	0.9955	51.138	0.9955
Boat	51.135	0.9963	51.125	0.9962	51.141	0.9963
Couple	51.142	0.9964	51.148	0.9964	51.135	0.9964
Elaine	51.138	0.9970	51.144	0.9970	51.131	0.9969
Grape	51.151	0.9965	51.147	0.9965	51.136	0.9965
Mandrill	51.133	0.9987	51.140	0.9987	51.133	0.9987
Peppers	51.155	0.9962	51.145	0.9962	51.149	0.9962
Splash	51.152	0.9950	51.148	0.9950	51.143	0.9950
Zelda	51.162	0.9958	51.145	0.9958	51.133	0.9957
Average	51.145	0.9964	51.143	0.9964	51.138	0.9964

(2.1) Use LSBC method to extract shared bits from B'_m of each stego image to form a configuration of the reverse LMCA, and then k initial configurations $C'^{(0)}, \ldots, C'^{(k-1)}$ can be retrieved.

(2.2) Calculate α evolutions of the reverse LMCA by (6) and gain $C'^{(\alpha)}, \ldots, C'^{(\alpha+k-1)}$, namely, the $C^{(k-1)}, \ldots, C^{(0)}$ of the original LMCA.

(2.3) Divide $C'^{(\alpha+k-1)}$ into multiple parts of bits and put them back to B'_m of the stego image. Use $C'^{(\alpha)}, \ldots, C'^{(\alpha+k-2)}$ to construct the mth pixel block of the secret image.

(3) After Step (2), the secret image can be recovered losslessly. If there are idle pixels in the stego images, then assign the idle pixels of one stego image to the cover image CI, which happens when not all the capacity of CI is used for secret data embedding. Then CI can also be restored losslessly.

4. Experimental Results and Comparisons

We will evaluate this scheme from different aspects such as the visual quality of stego images, embedding capacity, embedding ways, and other features. Comparisons will be made with the previous schemes that can restore both the cover image and the secret image losslessly. These comparisons will show that our scheme has many better features among the previous schemes. Two popular ways to evaluate the quality of stego images are peak-signal to noise rate (PSNR) and structural similarity (SSIM) [33]. PSNR is represented as follows:

$$\mathrm{PSNR} = 10 \log_{10} \left(\frac{255^2}{\mathrm{MSE}} \right) \mathrm{dB}. \tag{9}$$

The mean square error (MSE) of an image with $H \times W$ pixels is defined as

$$\mathrm{MSE} = \frac{1}{H \times W} \sum_{u=1}^{H} \sum_{v=1}^{W} \left(x_{uv} - y_{uv} \right)^2, \tag{10}$$

where x_{uv} is the original pixel value from the cover image and y_{uv} is the processed pixel value of a stego image.

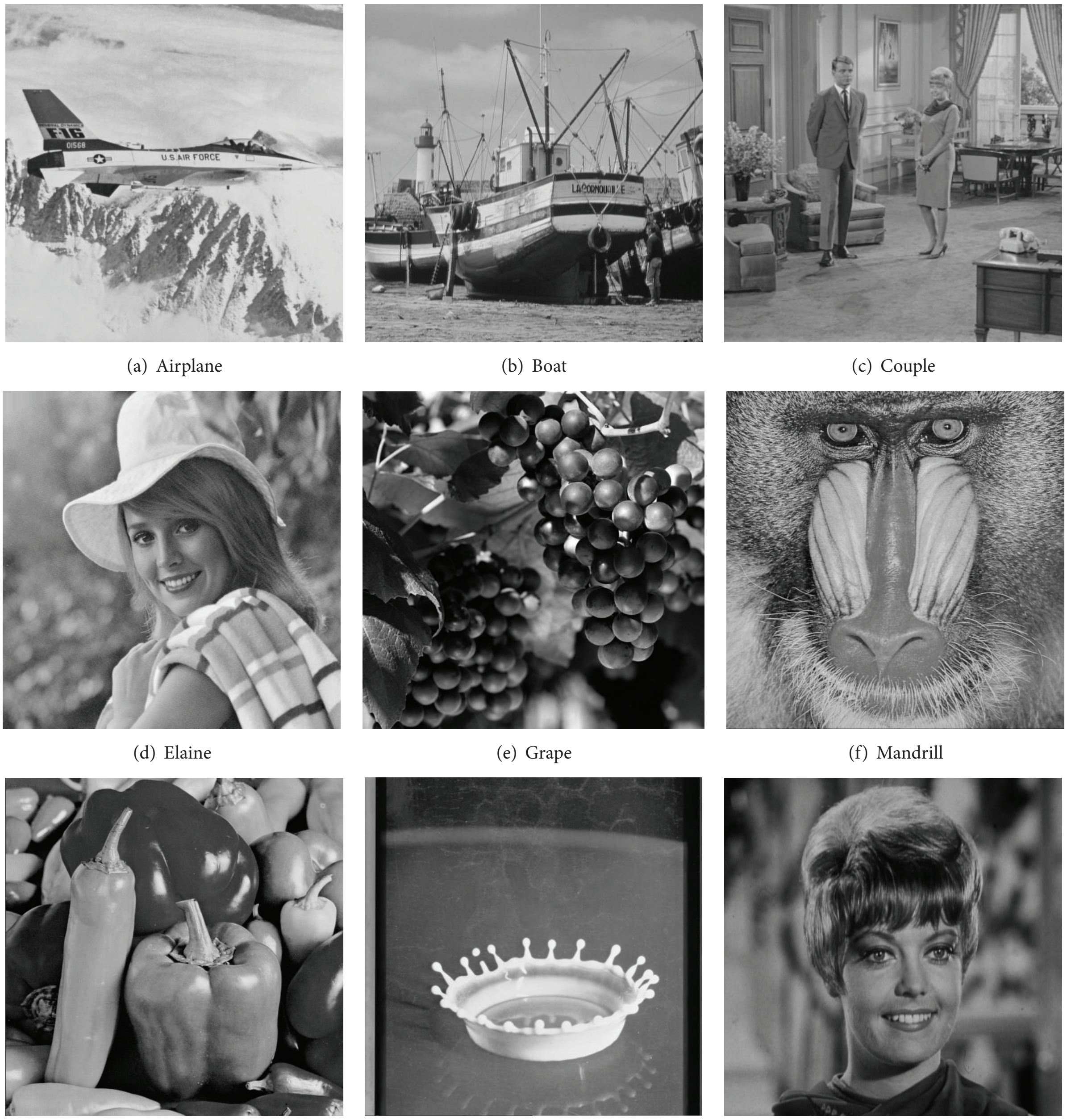
Figure 4: Secret image.

SSIM evaluates the quality of stego images in terms of human visual system and structural similarity. The big values of SSIM indicate that the fidelity of stego images is approximate to the original cover images [22]. For details about the way to get SSIM values, please refer to [33].

4.1. Evaluation on the Quality of Stego Images. The visual quality of stego images is related with the security of this scheme in some extent. When this quality is too low, stego images are easy to be suspected by vicious attackers.

Table 1 shows the quality values of the stego images generated from various cover images of Figure 5 and the secret image of Figure 4 with $k = 3$, $\mathrm{SIW} = \mathrm{SIH} = 256$, and $\mathrm{CIW} = \mathrm{CIH} = 512$. It is easy to know that $\mathrm{CIPNL} = \mathrm{CIPNH} = 8$. Therefore, eight pixels of CI are used to construct s_0 every time in sharing and embedding phase. Obviously, no more than one least significant bit of each pixel of CI will be changed. Hence, CI is modified slightly in sharing and embedding phase. The average PSNR values of stego image 1, stego image 2, and stego image 3 are 51.145, 51.143, and 51.138, respectively.

To describe the visual quality of stego images more precisely, Figure 6 contains enlarged partial areas of two cover



(a) Airplane (b) Boat (c) Couple

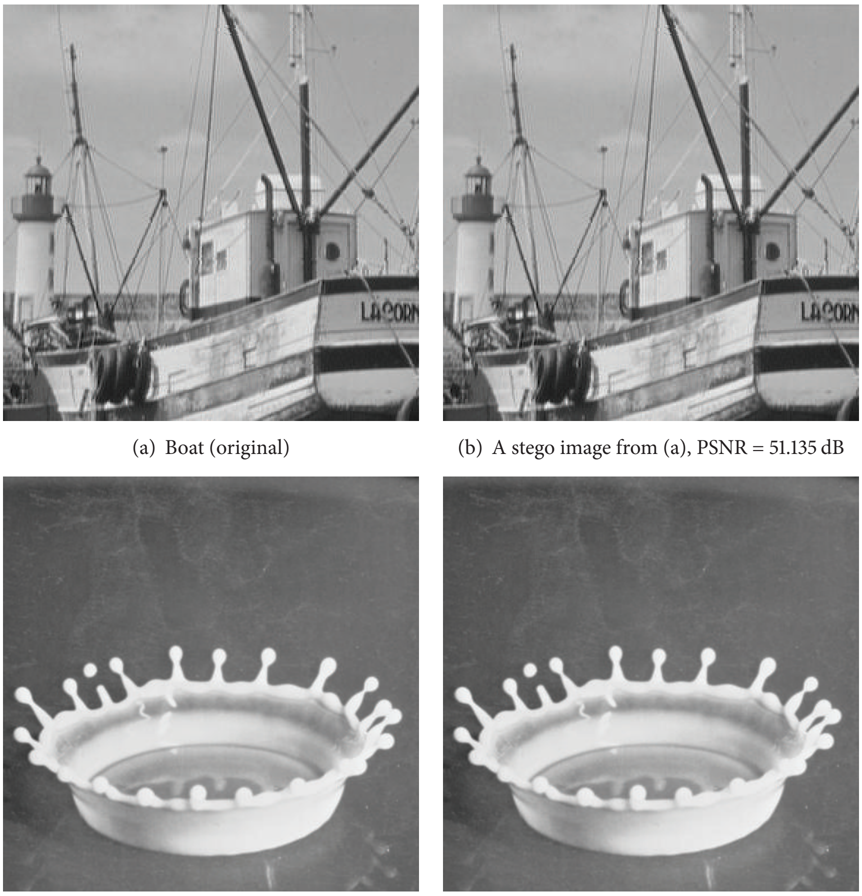
(d) Elaine (e) Grape (f) Mandrill

(g) Peppers (h) Splash (i) Zelda

Figure 5: Cover images.

images, and enlarged partial areas of their first stego images respectively. From visual perception, the difference between the enlarged part of one cover image and the corresponding part of its stego image is indistinguishable. Therefore, this scheme can embed a secret image into a cover image effectively and generate stego images with relatively high quality.

4.2. Analysis of the Embeddable Capacity, Embedding Ways, and Other Features. In this paper, one-dimensional memory cellular automata is used and each group of $(k - 1)$ secret pixels could be seen as being embedded into each eight-bit number s_0. If k increases (resp., decreases), more (resp., less) pixels will be embedded into s_0 each time. Hence, factor k affects the embeddable secret capacity. Suppose that the number of pixels to construct s_0 is h; if h becomes smaller, then s_0 will be constructed from fewer pixels and the value of each affected pixel of stego images will change more. Hence, the visual quality of stego images will become worse. Meanwhile, a smaller value of h brings larger embeddable secret capacity. On the contrary, a bigger value of h brings the better visual quality of stego images and smaller embeddable secret capacity. One can find that there is a tradeoff between the embeddable secret capacity and the visual quality of stego images. Therefore, factors k and h affect the embeddable



(a) Boat (original)

(b) A stego image from (a), PSNR = 51.135 dB

(c) Splash (original)

(d) A stego image from (c), PSNR = 51.152 dB

Figure 6: Comparison of enlarged partial areas of the cover images and the corresponding stego images.

capacity whose value is $(k-1)\times\lfloor(\mathrm{CIH}\times\mathrm{CIW})/h\rfloor$, $h\in[2,8]$. Let us define a secret image sharing scheme with steganography which can restore both the secret image and the cover image losslessly as a perfect reversible scheme. From Table 2, one can find that this scheme and [22] have the biggest value of maximum capacity among the perfect reversible schemes. The schemes of [26, 27] will have the same maximum capacity as this scheme and [22] only when $m\geq 16$. However, modulus operation often leads to overflow or other issues. Therefore, the schemes using LSBC can have the full maximum capacity..

To describe an embedding way, there are two factors, which are the method to conceal secret data into a cover image, namely, steganography method, and the method to spread secret data over a cover image, namely, embedding method. Table 2 shows the steganography methods used by different perfect reversible schemes. The threshold value k is 4 for [24] and 3 for other schemes in Table 2. Modulus operation is more popular while LSBC can make a scheme have a bigger value of maximum capacity and other better features. For embedding methods, we can have two definitions which are static embedding and dynamic embedding. Static embedding is the method to modify part of a cover image severely to embed secret data. However, dynamic embedding will consider the whole capacity and try to disperse secret data over all the cover image to make the whole extent of modifications to the cover image as low as possible. Besides, if part of a stego image is modified severely, it may attract attackers' attention. Hence, dynamic embedding can help improve the visual quality and security of stego images. Both this scheme and [22] use dynamic embedding. Besides, if all the schemes embed the same amount of secret data, the schemes with larger embedding capacity will generate stego images with better visual quality relatively. Therefore, these two schemes can generate stego images with better visual quality and security.

Furthermore, some other defects in the schemes using modulus operation need to be discussed here. Firstly, Wu et al. [27] pointed out that their scheme and the scheme of [26] cannot implement the (6,6)-threshold case directly. This is caused by the characteristics of modulus operation and the design of their schemes. If these two schemes want to implement the (6,6)-threshold case, they need extra storage space and special work. Secondly, the modulus value m is not independent in the schemes of [24, 27] but is affected by the threshold value k. For example, the schemes of [26, 27] have the same capacity with the same m, but they generate stego images with different visual quality while k equals 3. The reason is that m can be 3 in [26] and the optimal value of m is 16 in [27]. It is impossible for [27] to have m as 3 or

Table 2: Comparisons with previous perfect reversible schemes.

Schemes	Computation complexity	Functionality			PSNR (dB)	Maximum capacity (pixels)
		Steganographic method	Without extra expansion/storage	Dynamic embedding		
Lin et al. [24]	$O\left(n\log^2 n\right)$	Modulus operation	Yes	No	43.38	$\frac{\text{CIH} \times \text{CIW} \times (k-3)}{3}$
Chang et al. [25]	$O\left(n\log^2 n\right)$	Sudoku table	No	No	47.13	$\frac{\text{CIH} \times \text{CIW} \times (k-1)}{4}$
Lin and Chan [26]	$O\left(n\log^2 n\right)$	Modulus operation	Yes	No	47.81	$\frac{\text{CIH} \times \text{CIW} \times (k-1)}{\lceil \log_m 255 \rceil}$
Wu et al. [27]	$O(n)$	Modulus operation	Yes	No	37.99	$\frac{\text{CIH} \times \text{CIW} \times (k-1)}{\lceil \log_m 255 \rceil}$
Hu et al. [22]	$O\left(n\log^2 n\right)$	LSBC	Yes	Yes	51.86	$\frac{\text{CIH} \times \text{CIW} \times (k-1)}{2}$
Ours	$O(n)$	LSBC	Yes	Yes	51.15	$\frac{\text{CIH} \times \text{CIW} \times (k-1)}{2}$

Table 3: Comparisons of LSBC and modulus operation.

Methods	Features					
	Operation type	Computation cost	Without overflow issue	Utilize all pixels from a cover image	Affecting the selection of threshold values	Affected by the selection of threshold values
LSBC	Logical shift operation	Relatively low	Yes	Yes	No	No
Modulus operation	Modulus operation	Relatively high	No	No	Yes	Yes

any value less than 16 when k equals 3 and no extra special work is added. However, [26] can generate the stego images with the best visual quality when m equals 3. The smaller the value of m, the less the modification on each pixel of the cover image. If the capacity of one scheme is enough to embed a secret image, then it is best to let the value of m be as small as possible. Therefore, the visual quality of the stego images generated by [26] is much better than that by [27] even if they appear to have the same capacity. Moreover, m is not independent and its optimal value is 7 when k equals 4 in [24]. The capacity of [24] is enough to embed a secret image with the size 256 × 256 into a cover image with the size 512 × 512 when k equals 4, no matter what the value of m is. However, m cannot be 3 or 5 due to the design of the scheme and the characteristics of modulus operation. Hence, the visual quality of the stego images generated by the schemes of [22, 25, 26] and our scheme is much better than that by [24] even with a smaller value of k. Our scheme does not have the two defects above. From Table 2, we can see that our scheme can generate stego images with better visual quality than all the previous schemes of [24, 26, 27] using modulus operation. In order to know the differences between LSBC and modulus operation better, Table 3 shows the features of them. One can find that LSBC has better characteristics in many aspects.

Lastly, other interesting features of this scheme have also been studied. The first one is that this scheme is one of the fastest schemes that can restore both the secret image and the cover image losslessly in the world by far. Cellular automata have very low computation complexity and support parallel computing. This scheme can run faster when using parallel computing. In other words, this scheme can make better use of multicore computer after a slight modification. Besides, LSBC mainly uses logical shift operations and has lower computation complexity than that of modulus operation. The second one is that this scheme can resist differential attack. Cellular automata has the characteristic of resisting differential attack. Generally, if one small change in the original secret image causes a significant change in each generated share with respect to diffusion and confusion, then any differential attack is useless. Our scheme is sensitive with respect to minor changes in the original secret image. The third one is that this scheme can protect the secret image effectively if $k-1$ stego images are obtained by a hacker, or even if more than $k-1$ stego images are gained by him/her. For the latter case, if there are no k continuous stego images, this will not meet the requirement of cellular automata to conduct the reverse evolutions, which requires that at least k stego images for restoring the secret image should be continuous. Then, this hacker cannot get any information about the secret image. Generally, in recovering phase, any $k-1$ or fewer shadows cannot provide sufficient information to restore the original secret image. For the proof of this feature, please refer to Section 3.2 in the scheme of [31].

5. Conclusions

It is important to restore both the secret image and the cover image losslessly in some situations, such as the artistic, medical, and legal domains. There are already some excellent schemes that can achieve this purpose. The new scheme proposed in this paper avoids the defects existing in some previous perfect reversible schemes and has the better features of low computation complexity, resisting differential attack, parallel computing, relatively larger value of maximum capacity, better visual quality of stego images, dynamic embedding, and so on.

Notations

k:	The threshold of this scheme
n:	The number of the participants
r:	Radius of neighborhood for the LMCA
N:	Number of cells in a configuration of the LMCA
CI:	The cover image
SI:	The secret image
$P_1, \ldots, P_n$:	The participants
SH_i:	The shares to be distributed to the participant P_i
STI_i:	The stego image corresponding to P_i
CIW, CIH:	The width and height of CI, respectively
SIW, SIH:	The width and height of SI, respectively
s_0:	The number generated by LSBC and its length is 8 bits
CIPNL:	The size of a smaller pixel block of CI, and the block is used to construct s_0
CIPNH:	The size of a bigger pixel block of CI, and the block is used to construct s_0, CIPNL $\leq$ CIPNH.

Conflict of Interests

The authors declare that there is no conflict of interests regarding the publication of this paper.

Acknowledgment

This research was supported by the National Science Foundation of China under Grants nos. 61272173 and 61401060.

References

[1] A. Shamir, "How to share a secret," *Communications of the Association for Computing Machinery*, vol. 22, no. 11, pp. 612–613, 1979.

[2] G. R. Blakley, "Safeguarding cryptographic keys," in *Proceeding of the National Computer Conference*, vol. 48 of *AFIPS Conference Proceedings*, pp. 313–317, 1979.

[3] M. Naor and A. Shamir, "Visual cryptography," in *Cryptology-Eurocrypt '94*, vol. 950 of *Lecture Notes in Computer Science*, pp. 1–12, Springer, Berlin, Germany, 1995.

[4] S. Droste, "New results on visual cryptography," in *Advances in Cryptology—CRYPTO '96*, vol. 1109 of *Lecture Notes in Computer Science*, pp. 401–415, Springer, Berlin, Germany, 1996.

[5] C. Blundo, S. Cimato, and A. De Santis, "Visual cryptography schemes with optimal pixel expansion," *Theoretical Computer Science*, vol. 369, no. 1–3, pp. 169–182, 2006.

[6] C.-N. Yang, "New visual secret sharing schemes using probabilistic method," *Pattern Recognition Letters*, vol. 25, no. 4, pp. 481–494, 2004.

[7] T.-H. Chen and K.-H. Tsao, "Threshold visual secret sharing by random grids," *Journal of Systems and Software*, vol. 84, no. 7, pp. 1197–1208, 2011.

[8] T. Hofmeister, M. Krause, and H. U. Simon, "Contrast-optimal k out of n secret sharing schemes in visual cryptography," in *Computing and Combinatorics*, vol. 1276 of *Lecture Notes in Computer Science*, pp. 176–185, Springer, Berlin, Germany, 1997.

[9] C. Blundo, A. de Santis, and D. R. Stinson, "On the contrast in visual cryptography schemes," *Journal of Cryptology*, vol. 12, no. 4, pp. 261–289, 1999.

[10] C. Blundo, P. D'Arco, A. De Santis, and D. R. Stinson, "Contrast optimal threshold visual cryptography schemes," *SIAM Journal on Discrete Mathematics*, vol. 16, no. 2, pp. 224–261, 2003.

[11] M. Krause and H. U. Simon, "Determining the optimal contrast for secret sharing schemes in visual cryptography," *Combinatorics, Probability and Computing*, vol. 12, no. 3, pp. 285–299, 2003.

[12] C.-C. Lin and W.-H. Tsai, "Visual cryptography for gray-level images by dithering techniques," *Pattern Recognition Letters*, vol. 24, no. 1–3, pp. 349–358, 2003.

[13] Y.-C. Hou, "Visual cryptography for color images," *Pattern Recognition*, vol. 36, no. 7, pp. 1619–1629, 2003.

[14] C.-N. Yang and T.-S. Chen, "Colored visual cryptography scheme based on additive color mixing," *Pattern Recognition*, vol. 41, no. 10, pp. 3114–3129, 2008.

[15] C.-C. Lin and W.-H. Tsai, "Secret image sharing with steganography and authentication," *Journal of Systems and Software*, vol. 73, no. 3, pp. 405–414, 2004.

[16] Y.-S. Wu, C.-C. Thien, and J.-C. Lin, "Sharing and hiding secret images with size constraint," *Pattern Recognition*, vol. 37, no. 7, pp. 1377–1385, 2004.

[17] C.-C. Chang, C.-Y. Lin, and C.-S. Tseng, "Secret image hiding and sharing based on the (t, n) -threshold," *Fundamenta Informaticae*, vol. 76, no. 4, pp. 399–411, 2007.

[18] C.-C. Thien and J.-C. Lin, "Secret image sharing," *Computers & Graphics*, vol. 26, no. 5, pp. 765–770, 2002.

[19] R. Zhao, J.-J. Zhao, F. Dai, and F.-Q. Zhao, "A new image secret sharing scheme to identify cheaters," *Computer Standards & Interfaces*, vol. 31, no. 1, pp. 252–257, 2009.

[20] C.-C. Chang, Y.-P. Hsieh, and C.-H. Lin, "Sharing secrets in stego images with authentication," *Pattern Recognition*, vol. 41, no. 10, pp. 3130–3137, 2008.

[21] C.-N. Yang, T.-S. Chen, K. H. Yu, and C.-C. Wang, "Improvements of image sharing with steganography and authentication," *Journal of Systems and Software*, vol. 80, no. 7, pp. 1070–1076, 2007.

[22] W.-T. Hu, M.-C. Li, C. Guo, and Y.-Z. Ren, "Reversible secret image sharing with steganography and dynamic embedding," *Security and Communication Networks*, vol. 5, no. 11, pp. 1267–1276, 2012.

[23] P. Li, P.-J. Ma, X.-H. Su, and C.-N. Yang, "Improvements of a two-in-one image secret sharing scheme based on gray mixing model," *Journal of Visual Communication and Image Representation*, vol. 23, no. 3, pp. 441–453, 2012.

[24] P.-Y. Lin, J.-S. Lee, and C.-C. Chang, "Distortion-free secret image sharing mechanism using modulus operator," *Pattern Recognition*, vol. 42, no. 5, pp. 886–895, 2009.

[25] C.-C. Chang, P.-Y. Lin, Z. H. Wang, and M. C. Li, “A sudoku-based secret image sharing scheme with reversibility,” *Journal of Communications*, vol. 5, no. 1, pp. 5–12, 2010.

[26] P.-Y. Lin and C.-S. Chan, “Invertible secret image sharing with steganography,” *Pattern Recognition Letters*, vol. 31, no. 13, pp. 1887–1893, 2010.

[27] X. T. Wu, D. H. Ou, Q. M. Liang, and W. Sun, “A user-friendly secret image sharing scheme with reversible steganography based on cellular automata,” *The Journal of Systems and Software*, vol. 85, no. 8, pp. 1852–1863, 2012.

[28] A. Aho and J. Hopcroft, *Design and Analysis of Computer Algorithms*, Pearson Education India, 1974.

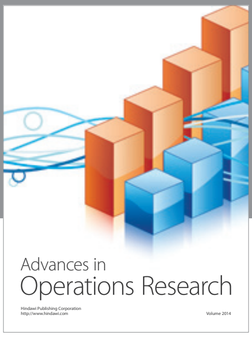
[29] Z. Eslami and J. Zarepour Ahmadabadi, “A verifiable multi-secret sharing scheme based on cellular automata,” *Information Sciences*, vol. 180, no. 15, pp. 2889–2894, 2010.

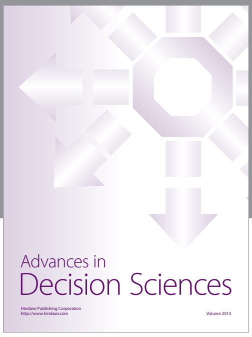
[30] T. Toffoli and N. H. Margolus, “Invertible cellular automata: a review,” *Physica D: Nonlinear Phenomena*, vol. 45, no. 1–3, pp. 229–253, 1990.

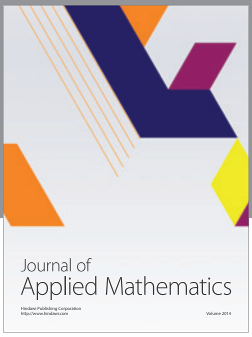
[31] A. Martin del Rey, J. P. Mateus, and G. R. Sanchez, “A secret sharing scheme based on cellular automata,” *Applied Mathematics and Computation*, vol. 170, no. 2, pp. 1356–1364, 2005.

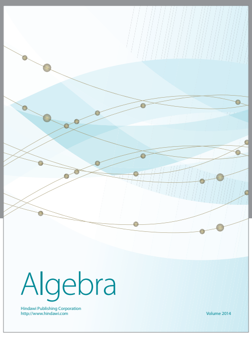
[32] A. Menezes, P. van Oorschot, and S. Vanstone, *Handbook of Applied Cryptography*, CRC Press, Boca Raton, Fla, USA, 1997.

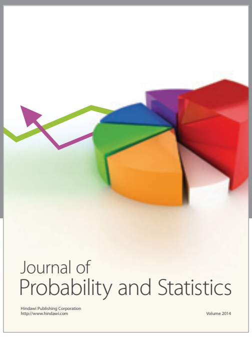
[33] Z. Wang, A. C. Bovik, H. R. Sheikh, and E. P. Simoncelli, “Image quality assessment: from error visibility to structural similarity,” *IEEE Transactions on Image Processing*, vol. 13, no. 4, pp. 600–612, 2004.


Advances in
Operations Research

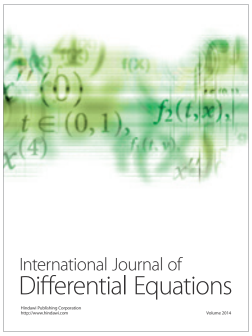

Advances in
Decision Sciences


Journal of
Applied Mathematics

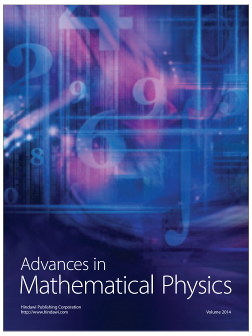

Algebra

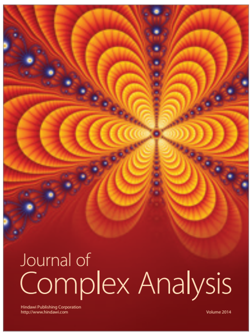

Journal of
Probability and Statistics

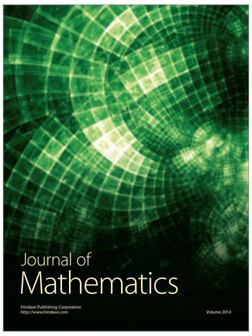
The Scientific
World Journal

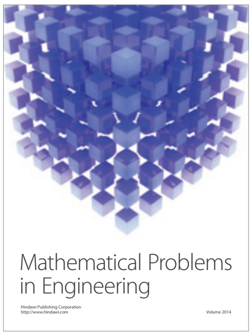

International Journal of
Differential Equations

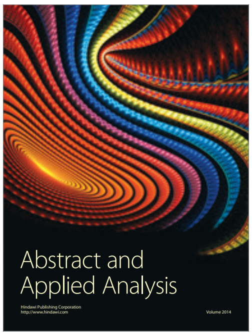
International Journal of
Combinatorics

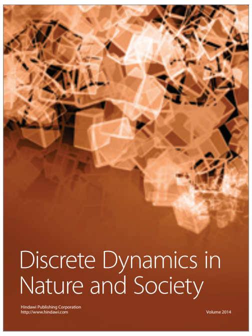

Advances in
Mathematical Physics

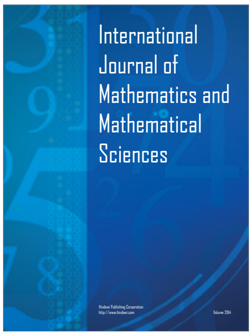

Journal of
Complex Analysis


Journal of
Mathematics


Mathematical Problems
in Engineering

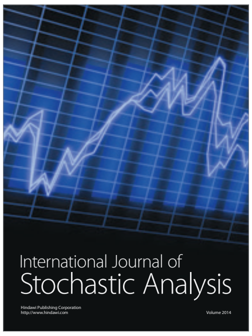

Abstract and
Applied Analysis


Discrete Dynamics in
Nature and Society


International
Journal of
Mathematics and
Mathematical
Sciences

Journal of
Discrete Mathematics

Journal of
Function Spaces


International Journal of
Stochastic Analysis

Journal of
Optimization