International Journal of Electronic Commerce Studies
Vol.7, No.2, pp.135-154, 2016
doi: 10.7903/ijecs.1467

SHARE A SECRET IMAGE WITH INVERTIBLE AND LOSSY CHARACTERISTICS

Chien-Chang Chen
Tamkang University
No.151, Yingzhuan Rd., Tamsui Dist., New Taipei City 251, Taiwan
ccchen34@mail.tku.edu.tw

Chong-An Liu
Tamkang University
No.151, Yingzhuan Rd., Tamsui Dist., New Taipei City 251, Taiwan
fty2058@gmail.com

Yu-Jing Lin
Tamkang University
No.151, Yingzhuan Rd., Tamsui Dist., New Taipei City 251, Taiwan
ariel_1225@hotmail.com.tw

ABSTRACT

Conventional (*t*, *n*) secret image sharing schemes share a secret image to *n* shared images, where any *t* shared images recovers the secret image. Among these shared images, noise-like properties easily draw attacker attention. Embedding shared images in meaningful cover images thus efficiently reduces attacker attention. This paper presents a different-expansion technique based invertible secret image sharing scheme that allows participants to perfectly restore the secret image and cover images. The proposed scheme also contains a lossy property which means that cover images do not have to be perfectly recovered to share larger secret images. The proposed scheme performs well with M-ary number systems, allowing users to determine the trade-off between covered image quality and secret image size. Experimental results show that the proposed scheme shares a large secret image and has good covered image quality.

Keywords: Secret Image Sharing, Different Expansion, Lossy Recovery

1. INTRODUCTION

Digital images are frequently shared, therefore, protecting valuable images during storage or transmission is important. Secret image sharing techniques are efficient methods of protecting digital images. This technique works by sharing a secret image among noise-like shared images and then gathering sufficient shared images to recover the secret image.

Thien and Lin first presented their Shamir-Lagrange based secret image sharing method[1]. Then, many researchers proposed other functional secret image sharing schemes, such as reduced load for sharing multiple secret images[2, 3], cheater identification[4], progressive sharing[5, 6], weighted sharing[7], visual cryptography and secret image sharing[8], authentication strength[9], and scalable sharing[10].

In these conventional secret image sharing schemes, shared images always resemble noise images. This noise-like property attracts attacker attention and increases efforts to maintain their visual similarity. Embedding shared images to meaningful images is a feasible method of eliminating these disadvantages. Chang et al.[2] used a steganography technique and Chinese Remainder Theorem to share and authenticate images. Lin and Tsai[11] presented steganography and parity check techniques to share and authenticate images. Lin et al.[12] used pixels in secret and cover images to apply to the Shamir sharing function for pixel calculation of shared images. Lin and Chan[13] improved this technique[12] to increase the embedded capacity. Because an invertible secret image sharing scheme requires embedding abilities, the reversible watermarking method efficiently solves this problem; thus the proposed scheme used a reversible watermarking scheme to create an efficient invertible secret image sharing scheme. Difference expansion methods[14, 15] can reversibly embed watermarks into cover images. Hu et al.[16] improved embedding performance using M-ary number systems. Some other techniques, such as cellular automata[17] and Boolean operations[18, 19, 20], are adopted for their efficiency on sharing secret images.

This paper presents an M-ary number based invertible secret image sharing scheme. The secret image was first shared to the shared images using an M-ary number system. The shared images were then embedded into cover images using M-ary number reversible image watermarking. M-ary sharing and M-ary reversible watermarking techniques are important to the proposed scheme. The proposed scheme involves invertible or lossy methods. The lossy method enables larger secret images than the invertible method.

This paper is organized as follows. Section 2 reviews the relevant literature. Section 3 introduces the proposed invertible secret image sharing scheme. Sections 3.1 and 3.2 present the algorithms that share a secret image to cover images and recover the secret image from covered images, respectively. Section 4 presents the experimental results and compares the proposed scheme with other methods. Section 5 provides a conclusion and future research suggestions.

2. LITERATURE REVIEW

This section reviews the relevant literature used to develop the proposed scheme. Hu et al.[16] presented a difference expansion improved scheme that requires an extra load to increase embedded image quality. The conventional difference expansion method[15] embeds a watermark bit w to pixel pairs (s_1, s_2) to acquire embedded pair (s_1', s_2') using Eq. (1).

$$\begin{gathered} d = s_1 - s_2, \quad d' = 2 \times d + w \\ m = \left\lfloor \frac{\lvert s_1 + s_2 \rvert}{2} \right\rfloor \\ s_1' = m + \left\lfloor \frac{\lvert d' + 1 \rvert}{2} \right\rfloor, \quad s_2' = m - \left\lfloor \frac{\lvert d' \rvert}{2} \right\rfloor \end{gathered} \tag{1}$$

Because the value of d determines the embedded quality, Hu et al. used exponent computation and an extra location map (LM) to reduce the difference d requires. This method requires an extra LM to record the difference type. The method reduces pixel difference d from two pixels s_1 and s_2 to acquire new difference d' using Eq. (2).

$$d' = \begin{cases} d, & \text{if } d < 2 \\ d - 2^{\lfloor \log_2 d \rfloor - 1}, & \text{if } 2 \times 2^{\lfloor \log_2 d \rfloor - 1} \le d \le 3 \times 2^{\lfloor \log_2 d \rfloor - 1} - 1 \\ d - 2^{\lfloor \log_2 d \rfloor}, & \text{if } 3 \times 2^{\lfloor \log_2 d \rfloor - 1} \le d \le 4 \times 2^{\lfloor \log_2 d \rfloor - 1} - 1 \end{cases} \tag{2}$$

The LM for the difference type is calculated using Eq. (3).

$$\mathrm{LM} = \begin{cases} 0, & \text{if } 2 \times 2^{\lfloor \log_2 d \rfloor - 1} \le d \le 3 \times 2^{\lfloor \log_2 d \rfloor - 1} - 1 \\ 1, & \text{if } 3 \times 2^{\lfloor \log_2 d \rfloor - 1} \le d \le 4 \times 2^{\lfloor \log_2 d \rfloor - 1} - 1 \end{cases} \tag{3}$$

When extracting the watermarks, original difference d is calculated using Eq. (4).

$$d = \begin{cases} d' + 2^{\lfloor \log_2 d' \rfloor}, & if \quad \mathrm{LM} = 0 \\ d' + 2^{\lfloor \log_2 d' \rfloor + 1}, & if \quad \mathrm{LM} = 1 \end{cases} \tag{4}$$

This method used an extra LM to reduce the difference between a pair of pixels. Distortion after watermark embedding can be then reduced.

3. PROPOSED INVERTIBLE SECRET IMAGE SHARING SCHEME

The proposed scheme includes two types of algorithms: sharing and recovery. Each algorithm requires two procedures. The sharing algorithm involves sharing and embedding procedures. The sharing procedure shares a secret image to M-ary shared images. Each M-ary shared image is then embedded into a cover image, producing a covered image in the embedding procedure.

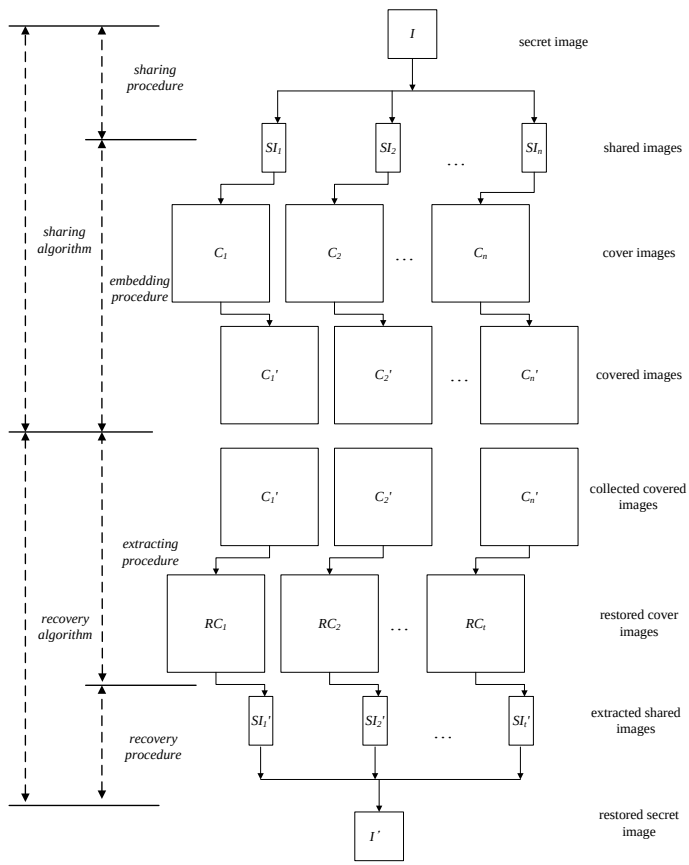

Figure 1. Structure of the proposed scheme

The recovery algorithm involves extracting and recovering procedures. The extracting procedure extracts M-ary shared images from covered images. The recovery procedure then uses these extracted M-ary shared images to recover the secret image. The entire process is performed using M-ary digits, allowing users to determine the number m to control the quantity in the covered image. Figure 1 shows the structure of the proposed scheme, including sharing and recovery algorithms.

3.1 Sharing Algorithm

This section describes the sharing algorithm—including the sharing and embedding procedures—of the proposed invertible secret image sharing scheme. As shown in Figure 1, the secret image is shared to shared images SI_k (k = 1, 2, …, n) in the sharing procedure. In the embedding procedure, each SI_k is embedded into cover image C_k to acquire covered image C_k'.

3.1.1 Sharing Procedure

The sharing procedure shares the secret image to n shared images SI_k(k=1, 2, …, n). Each pixel in the secret image is first translated to M-ary digit numbers. An $H \times W$ secret image acquires numbers using s_j $(j = 1,2,..., H \times W \times \lceil \log_m 255 \rceil)$, where m is a pre-determined prime number. The conventional Shamir method is then used to acquire shared images. Figure 2 shows the sharing procedure. Algorithm of the sharing procedure is described as follow:

1. Translate the $H \times W$ secret image to M-ary numbers m_j $(j = 1,2,..., H \times W \times \lceil \log_m 255 \rceil)$.

2. Partition numbers m_j to $\frac{H \times W \times \lceil \log_m 255 \rceil}{t}$ sets of t numbers and apply the following steps to each set of t numbers:

2.1 Assign a set of t numbers to parameters s_0, s_1, …, s_{t-1} in Eq. (5)

$$F(x) = (s_0 + s_1 x^1 + ... + s_{t-1} x^{t-1}) \bmod 251 \quad (5)$$

2.2 Calculate shared number y_j using participant secret key k_j and Eq. (5)

$$y_j = F(k_j) \quad (6)$$

3. Collect all y_j to form shared image SI_j for participant j.

A prime number is needed in conventional Shamir–Lagrange method and the number is determined by 251 in Step 2.1. Therefore, all parameters s_i in Step 2 must be restricted between 0 and 250. However, largest pixel value is 255 in digital images. Consequently, this gap can be solved by Thien and Lin's method [1]. Their method splits a pixel number to 250 and the remainder when the pixel is larger than 249.

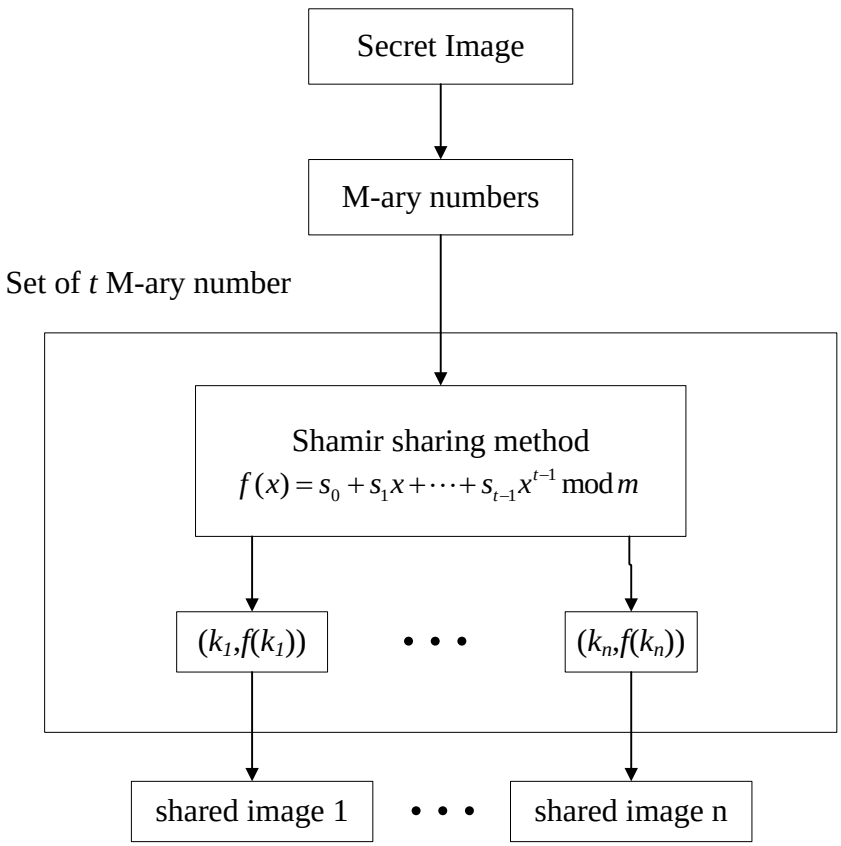

Figure 2. Steps of sharing procedure

3.1.2 Embedding Procedure

In the embedding procedure, each shared image SI_j is embedded into its corresponding cover image, C_j. The applied embedding strategy improves Hu et al.[16] method of embedding using Eq. (7) and recording the LM as calculated in Eq. (8).

$$d' = \begin{cases} 0, & \text{if } d = 0 \\ d - 2^{\lfloor \log_2 d \rfloor - 1}, & \text{if } 2 \times 2^{\lfloor \log_2 d \rfloor - 1} \le d \le 3 \times 2^{\lfloor \log_2 d \rfloor - 1} - 1 \\ d - 2^{\lfloor \log_2 d \rfloor}, & \text{if } 3 \times 2^{\lfloor \log_2 d \rfloor - 1} \le d \le 4 \times 2^{\lfloor \log_2 d \rfloor - 1} - 1 \end{cases} \tag{7}$$

$$\mathrm{LM} = \begin{cases} 0, & \text{if } d = 0 \\ 1, & \text{if } d = 1 \\ 0, & \text{if } 2 \times 2^{\lfloor \log_2 d \rfloor - 1} \le d \le 3 \times 2^{\lfloor \log_2 d \rfloor - 1} - 1 \\ 1, & \text{if } 3 \times 2^{\lfloor \log_2 d \rfloor - 1} \le d \le 4 \times 2^{\lfloor \log_2 d \rfloor - 1} - 1 \end{cases} \tag{8}$$

The shared image is the watermarks to embed into the differences between each set of two numbers in a cover image. Therefore, pixels in cover image C_j are translated to M-ary numbers and then partitioned to groups of two numbers for embedding watermarks extracted from shared image SI_j. Figure 3 shows the proposed embedding procedure steps. The algorithm for embedding shared image SI_j to cover image C_j is described as follows:

1. Translate cover image C_j to M-ary numbers. p_k represents the translated M-ary numbers.

2. Partition p_k to sets of two M-ary numbers and apply the following steps to each set of two M-ary numbers:

2.1 Let the two M-ary numbers be (p_t, p_{t+1}).

2.2 Replace least significant M-ary number p_t with y producing new pair (p'_t, p'_{t+1}), where y is extracted from shared image SI_j in sharing procedure. Store the replaced least significant M-ary number to data stream A.

2.3 Use the proposed difference expansion method to embed an M-ary number to (p'_t, p'_{t+1}) using following steps:

2.3.1 Translate (p'_t, p'_{t+1}) to decimal numbers and calculate the mean l and difference d of (p'_t, p'_{t+1}).

$$l = \left\lfloor \frac{(p'_t + p'_{t+1})}{2} \right\rfloor \qquad d = p'_k - p'_{k+1} \tag{9}$$

2.3.2 Use Eqs. (7) and (8) to acquire reduced difference *d'* and *LM* bit.

2.3.3 Embed an M-ary number y to difference d using $d'=m\times d+y$, where y is extracted from shared image SI_j in the sharing procedure.

2.3.4 Recover the two embedded M-ary numbers (p_t'', p_{t+1}'') from mean l and new difference d'.

$$p_t'' = l + \left\lfloor \frac{|d'+1|}{2} \right\rfloor \qquad (10)$$

$$p_{k+1}'' = l - \left\lfloor \frac{|d'|}{2} \right\rfloor$$

2.3.5 Store new pair (p_t'', p_{t+1}'') as a pair of pixels in covered image C_j'.

3. Repeat these steps and then recover the M-ary numbers to image pixels to obtain covered image C_j'.

Data stream A and *LM* generated in Steps 2.2 and 2.3.2, respectively, also must be stored. Only the least significant replacement calculates under M-ary number in Step 2. All other steps work under the decimal number.

3.2 Recovery Algorithm

The secret image can be recovered using the recovery algorithm, which includes extracting and recovery procedures, as shown in Fig. 1. Restored shared images are extracted from covered images in the extracting procedure, and t restored shared images $RC_j(j=1,2,\ldots,t)$ reconstruct the secret image in the recovery procedure.

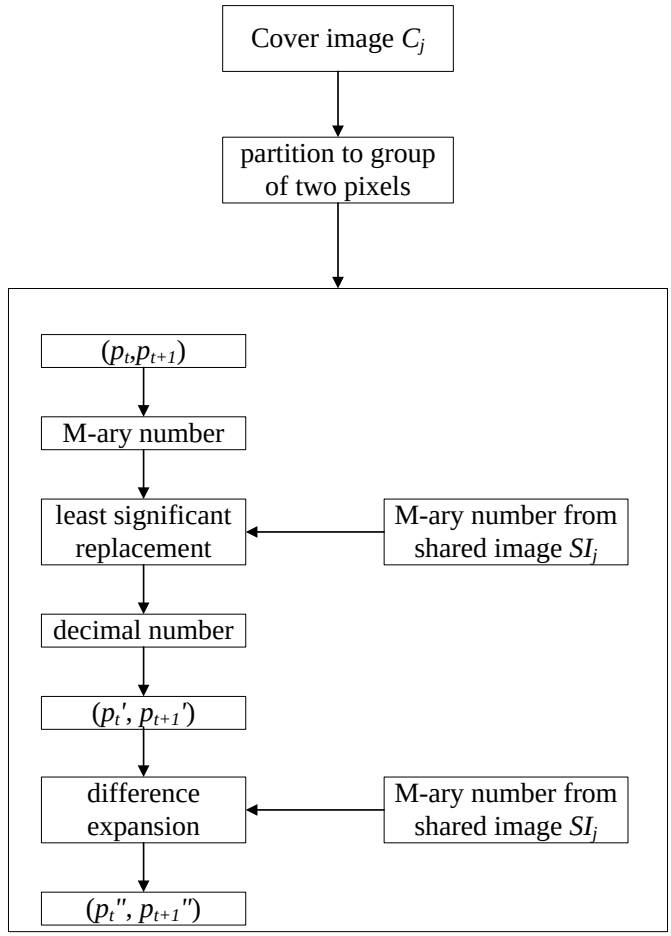

Figure 3. Steps of embedding procedure

3.2.1 Extracting Procedure

The extracting procedure requires t covered images, represented by C_j' (j=1,2,…,t). The extracting procedure extracts shared image SI_j' and restores cover image RC_j from covered image C_j'. The extracting procedure is described as follows:

1. Apply the following steps to every collected covered image C_j' (*j=1,2,…,t*) to extract covered image RC_j from each pair of pixels.

1.1 Assume that (p_k, p_{k+1}) denotes a pair of pixels in the covered image.

1.2 Calculate mean l and difference d using Eq. (11).

$$l = \left\lfloor \frac{p_k + p_{k+1}}{2} \right\rfloor , \quad d = p_k - p_{k+1} \tag{11}$$

1.3 Translate d to M-ary numbers and extract least significant M-ary number b_1.

1.4 Acquire new difference d' from difference d using Eq. (12)

$$d' = \left\lfloor \frac{d}{m} \right\rfloor \tag{12}$$

where m is the selected M-ary number.

1.5 Recover original difference d'' from d' and the corresponding LM bit using Eq. (13).

$$d'' = \begin{cases} 0, & if \ d' = 0 \text{ and } LM = 0 \\ 1, & if \ d' = 0 \text{ and } LM = 1 \\ d' + u \times m^{\lfloor \log_m d' \rfloor - 1}, & if \ d' > 0 \text{ and } LM = 0 \\ d' + u \times m^{\lfloor \log_m d' \rfloor}, & if \ d' > 0 \text{ and } LM = 1 \end{cases} \tag{13}$$

1.6 Acquire original pair of pixels (p'_k, p'_{k+1}) using l and d''.

$$p'_k = l + \left\lfloor \frac{d'' + 1}{2} \right\rfloor , \quad p'_{k+1} = l - \left\lfloor \frac{d''}{2} \right\rfloor \tag{14}$$

1.7 Extract least significant M-ary number b_2 from pixel p'_k and replace it with the M-ary number extracted from data stream A to obtain the original pair of cover image pixels.

2. Combine extracted least significant numbers b_1 (Step 1.3) and b_2 (Step 1.7) to form extracted shared image SI'_j from covered image C'_j.

3. Gather the extracted pair in Step 1.7 to form original cover image RC_j.

Figure 4 shows the data required for invertible or lossy recovery of the cover image. Covered image C_j' and *LM* can only produce the roughly covered image. Data stream A is required to acquire the original cover image losslessly.

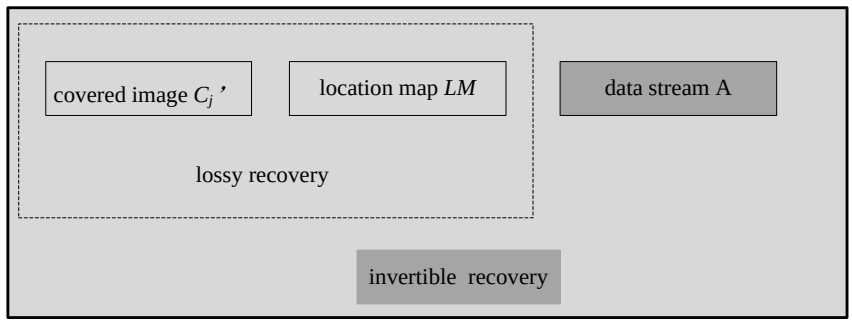

Figure 4. Data requirement between invertible and lossy recovery on cover images

3.2.2 Recovery Procedure

In the extracting procedure, the *t* extracted shared images SI_j' (*j=1,2,…,t*) are acquired to recover the secret image. Applying SI_j' to the following recovery procedure (constructed by Lagrange interpolation) acquires the secret image.

1. Apply *t* extracted shared images SI_1', SI_2', …, SI_t' to the following calculation to acquire the secret image:

   1.1 Assume that x_1 , x_2 , …, x_t are secret keys for each shared image and $y_{i,j}$ representing pixels of SI_i' S_i with $1 \le j \le$ shared image size.

   1.2 For each set of $y_{i,j}$, apply $y_{i,j}$ to Eq.(15) to acquire b_o, b_1, …, b_{t-1}, which are the pixels in the secret image.

$$f_k(x) = \sum_{i=1}^{t} y_{i,j} \prod_{j=1, j\neq i}^{t} \frac{x - x_j}{x_i - x_j} \bmod 251 = (b_0 + b_1 x + \cdots + b_{t-1} x^{t-1}) \bmod 251 \quad (15)$$

2. Gather each set of b_i($i = 0,1,\ldots,t-1$) from Step 1.2 to obtain the secret image.

4. EXPERIMENTAL RESULTS AND DISCUSSION

This section discusses the experimental results of the proposed scheme. Figure 5 shows secret image, Jet, which is 256×256 in size. Figure 6 shows six test images, which are 512×512 in size. The determined thresholds (*t*, *n*) used were (2, 3).

Figure 5. The secret image: Jet

4.1 Experimental Results

This section discusses the effects of the proposed scheme on covered image quality and embedded capacity for different *m* values (*m* represents the selected M-ary number). Table 1 shows the covered image quality of different cover images with *m* = 7. The covered image maintains a PSNR quality of more than 43 dB. Two properties determine covered image quality. First, a cover image embeds different shared images to acquire nearly all covered image qualities. This property indicates that the proposed scheme exhibits a stable covered image quality. For example, covered images of test image Lena had a PSNR between 45 and 46 dB. Covered images of test image Cameraman had a PSNR between 44.61 and 45.49 dB. Second, covered image quality is proportional to cover the image structure. A cover image with a more complex structure has a worse covered image quality. For example, because of its highly complex structure, covered images of test images Baboon had a PSNR of almost 43 dB, which is lower than other covered images.

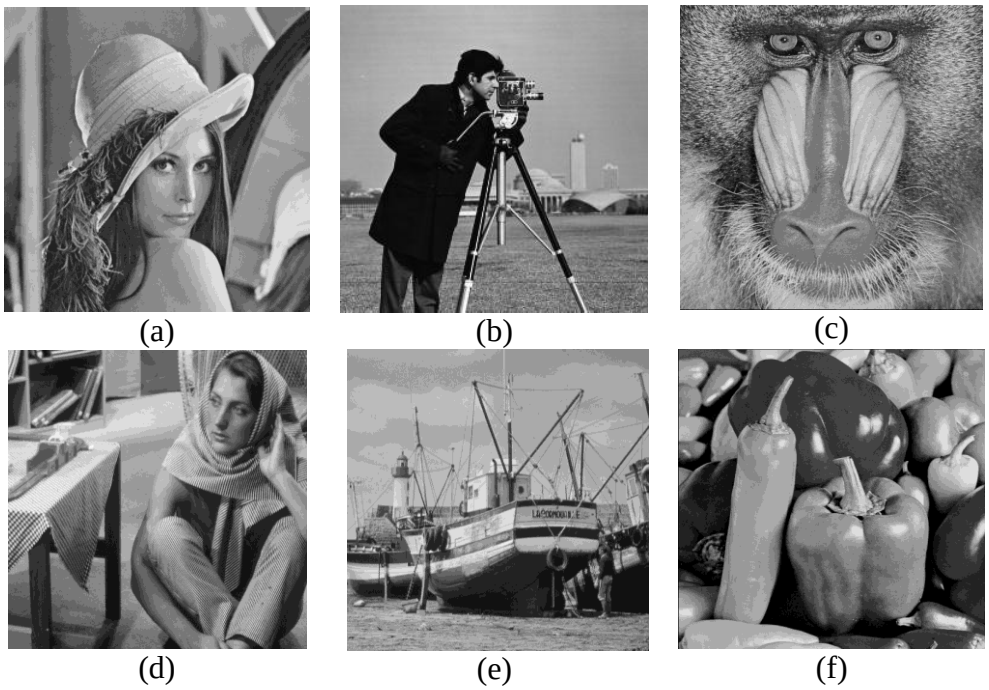

Figure 6. Test images: (a) Lena, (b) Cameraman, (c) Baboon, (d) Barbara, (e) Boat, (f) Peppers

Table 1. Covered image quality (m = 7)

cover image	Quality of covered image(PSNR)		
	covered image 1	covered image 2	covered image 3
Lena	45.78	46.01	45.06
Cameraman	45.31	45.49	44.61
Baboon	43.16	43.33	43.34
Barbara	45.47	45.83	45.19
Boat	45.52	45.92	45.10
Peppers	45.54	45.94	45.23

Table 2 shows M-ary image size and covered image quality at different m values.

Because the secret image is LENA (512×512), the M-ary image size is calculated using $512 \times 512 \times \frac{\lceil \log_m 255 \rceil}{8}$, where $\lceil \log_m 255 \rceil$ is the number of M-ary numbers required to indicate a pixel. For example, the

assignment of m=3 acquires $512 \times 512 \times \frac{\lceil \log_3 255 \rceil}{8} = 196608$ 3-ary numbers, and the covered image quality is 46.55 dB. By contrast, the assignment of m=19 acquires $512 \times 512 \times \frac{\lceil \log_{19} 255 \rceil}{8} = 65536$ 19-ary numbers, and the covered image quality is only 38.15 dB. This difference between covered images quality occurs because embedding a 19-ary number creates more image distortion than embedding a 3-ary number, even though the quantity of embedding a 19-ary number is less than embedding for a 3-ary number. Table 2 also lists the corresponding covered image quality, demonstrating the trade-off between m and covered image quality. Because small quantities cause covered image quality changes, the value listed in Table2 is the PSNR mean from the three covered images.

Table 2. *M*-ary image size v.s. covered image quality under different *m* selection

(512×512 Lena image)

M	*m*-ary image size (*m*-ary number)	Covered image quality (PSNR)
3	196608	46.55
5	131072	46.18
7	98304	45.62
11	98304	42.14
13	98304	40.51
17	65536	39.43
19	65536	38.15
23	65536	36.56
29	65536	35.31

The maximal embedded capacity, defined as the M-ary number, for a cover image with size $M \times N$ using (t, n) thresholds is $\frac{t \times H \times W}{\lceil \log_m 255 \rceil}$. Table 3 shows the maximal embedded capacity and quality of the covered image at different m values. A larger m value leads to more quantity embedded in a pixel, reducing the covered image quality. Therefore, a smaller m value

(such as 3) has the least embedded capacity, but the best image distortion. A larger *m* value (such as 17) has a good embedded capacity of 262144 bits, but a PSNR value of only 33.3 dB. Although *m* = 17 and 29 have the same embedded M-ary numbers, embedded 19-ary numbers lead to less image distortion than embedded 29-ary numbers.

Table 3. Most embedded capacity v.s. covered image quality under different *m* selection

(512×512 Lena image)

M	most embedded capacity (m-ary numbers)	covered image quality (PSNR)
3	87381	44.99
5	131072	43.57
7	174762	41.24
11	174762	37.87
13	174762	36.22
17	262144	33.30
19	262144	31.96
23	262144	30.60
29	262144	29.14

4.2 Discussion

This section compares the proposed scheme with two well-known invertible secret image sharing schemes[12, 13]. The three schemes preserve the following properties (also shown in Table 4):

1. The secret image is perfectly recovered.

2. The secret image is shared and embedded into cover images.

3. The cover image is perfectly recovered from covered images.

Table 4 compares the theoretical embedded capacity of the different schemes, based on image height *H*, width *W*, threshold *t*, and M-ary number system *m*. Number system *m* was set at 7. Figure 7 compares the capacity of the methods. It shows that the proposed scheme preserves a higher capacity than the other methods. The proposed scheme requires an

extra load. This is obtained by recording the least significant M-ary number and *LM* bits, which correspond with $\frac{H \times W}{\lceil \log_m 255 \rceil}$ and $\frac{H \times W}{16}$, respectively. Figure 8 shows pure capacity, defined by capacity minus extra storage. The proposed scheme performs almost as well as the Lin and Chan method [13]. The proposed scheme has nearly the same capacity as the other two methods but preserves better covered image quality with a higher PSNR. Moreover, the extra storage can be publicly announced for reducing overheads. The proposed scheme requires extra storage to show an invertible secret image sharing scheme. However, if the covered image does not have to be recovered to the original cover image quality, the extra storage can be further reduced to $\frac{H \times W}{16}$. Figure 9 shows pure capacity with lossy recovery. This means that the covered images were not recovered to the original cover image quality.

Table 4. Comparisons between the proposed scheme and important invertible secret image sharing schemes[12, 13] (test image: Lena)

items	Lin et al.[12]	Lin and Chan[13]	Proposed Scheme
losslessly recover cover image	yes	yes	yes
share to cover images	yes	yes	yes
recover the secret image perfectly	yes	yes	yes
capacity (pixels)	$\frac{(t-3) \times H \times W}{3}$	$\frac{(t-1) \times H \times W}{\lceil \log_m 255 \rceil}$	$\frac{t \times H \times W}{\lceil \log_m 255 \rceil}$
extra storage (bytes)	no	No	$\frac{H \times W}{\lceil \log_m 255 \rceil} + \frac{H \times W}{16}$
covered image quality	43 dB	42 dB	45.62 dB (m=7)

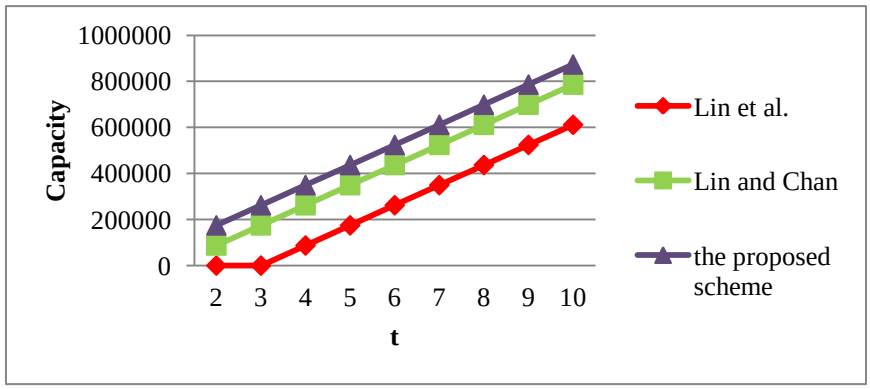

Figure 7. Capacity between the proposed scheme (m = 7) and important invertible secret image sharing schemes

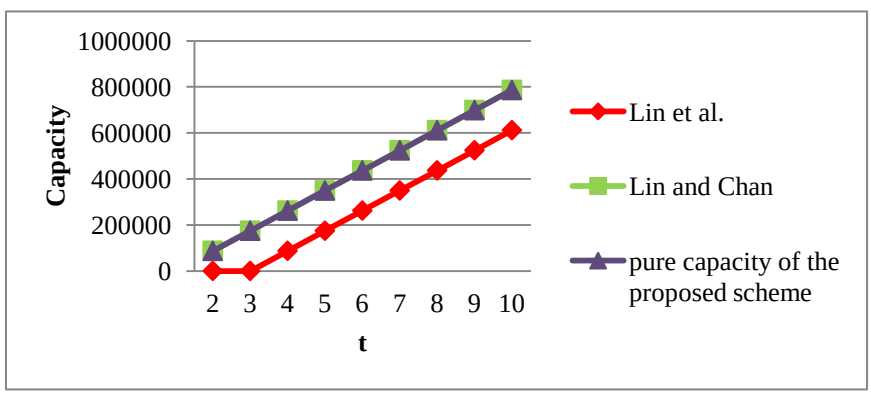

Figure 8. Pure capacity between the proposed scheme (m = 7) and important invertible secret image sharing schemes

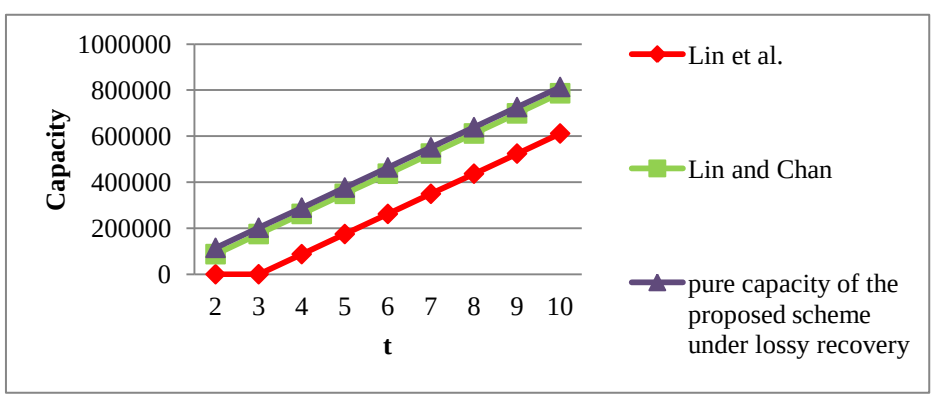

Figure 9. Pure capacity between the proposed scheme (m = 7 and lossy recovery) and important invertible secret image sharing schemes

5. CONCLUSION

This paper presents an invertible secret image sharing scheme. The proposed scheme uses the Shamir–Lagrange secret sharing method, a difference expansion method, and least significant M-ary number replacement to share a secret image and reversibly embedded shared images into cover images. Pre-determined parameter m controls the trade-off between capacity and covered image quality. Theoretical comparison and experimental results show that the proposed scheme has two important properties: it is invertible and has better capacity than two well-known schemes, and it has lossy property; therefore, if cover images are not recovered, capacity is further increased. This invertible, lossy, two-function secret image sharing scheme is efficient and high capacity. Merging with other functions merits our future study.

6. ACKNOWLEDGMENTS

This paper was partially supported by the National Science Council of the Republic of China under contract NSC 103-2221-E-032-051.

7. REFERENCES

[1] C.C. Thien, and J.C. Lin, Secret image sharing. *Computers and Graphics*, 26(5), 765-770, 2002. http://dx.doi.org/10.1016/S0097-8493(02)00131-0.

[2] C.C. Chang, Y.P. Hsieh, and C.H. Lin, Sharing secrets in stego images with authentication. *Pattern Recognition*, 41(10), 3130-3137, 2008. http://dx.doi.org/10.1016/j.patcog.2008.04.006.

[3] C.C. Chen, and W.J. Wu, A secure Boolean-based multi-secret image sharing scheme. *The Journal of Systems and Software*, 92, 107-114, 2014. http://dx.doi.org/10.1016/j.jss.2014.01.001.

[4] C.C. Chen, and C.A. Liu, Tamper-proof secret image sharing scheme for identifying cheated secret keys and shared images. *Journal of Electronic Imaging*, 22(1) 013008, 2013. http://dx.doi.org/10.1117/1.JEI.22.1.013008.

[5] W.P. Fang, Friendly progressive visual secret sharing. *Pattern Recognition*, 41(4), 1410-1414, 2008. http://dx.doi.org/10.1016/j.patcog.2007.09.004.

[6] C.P. Huang, C.H. Hsieh, and P.S. Huang, Progressive sharing for a secret image. *Journal of Systems and Software*, 83(3), 517-527, 2010. http://dx.doi.org/10.1016/j.jss.2009.10.012.

[7] S.J. Lin, L.S. Chen, and J.C. Lin, Fast-weighted secret image sharing. *Optical Engineering*, 48(7), 077008, 2009. http://dx.doi.org/10.1117/1.3168644.

[8] C.N. Yang, and C.B. Ciou, Image secret sharing method with two-decoding-options: Lossless recovery and previewing capability. *Image and Vision Computing*, 28(10), 1600-1610, 2010. http://dx.doi.org/10.1016/j.imavis.2010.04.003

[9] G. Ulutas, M. Ulutas, and V. Nabiyev, Secret image sharing scheme with adaptive authentication strength. *Pattern Recognition* Letters 34(3), 283-291, 2013. http://dx.doi.org/10.1016/j.patrec.2012.10.017.

[10] R.Z. Wang, and C.H. Su, Secret image sharing with smaller shadow images. *Pattern Recognition Letters*, 27(6), 551-555, 2006. http://dx.doi.org/10.1016/j.patrec.2005.09.021.

[11] C.C. Lin, and W.H. Tsai, Secret image sharing with steganography and authentication. *The Journal of Systems and Software*, 73(3), 405-414, 2004. http://dx.doi.org/10.1016/S0164-1212(03)00239-5

[12] P.Y. Lin, J.S. Lee, and C.C. Chang, Distortion-free secret image sharing mechanism using modulus operator. *Pattern Recognition*, 42(5), 886-895, 2009. http://dx.doi.org/10.1016/j.patcog.2008.09.014.

[13] P.Y. Lin, and C.S. Chan, Invertible secret image sharing with steganography. *Pattern Recognition Letters*, 31(13), 1887-1893, 2010. http://dx.doi.org/10.1016/j.patrec.2010.01.019.

[14] C.L. Liu, D.C. Lou, and C.C. Lee, Reversible data embedding using reduced difference expansion. In B.-Y. Liao, J.-S. Pan, L,C. Jain, M, Liao, H, Noda, and Anthony T. S. Ho (Eds.), *Proceedings of the Third International Conference on Intelligent Information Hiding and*

Multimedia Signal Processing (p430-436). Kaohsiung, Taiwan: IEEE Press, 2007. http://dx.doi.org/10.1109/IIH-MSP.2007.267.

[15] J. Tian, Reversible data embedding using a difference expansion, *IEEE Transactions Circuits Systems for Video Technology*, 13(8), 890-896, 2003. http://dx.doi.org/10.1109/TCSVT.2003.815962.

[16] Y. Hu, W. Song, and J. Hou, Improved reduced difference expansion based reversible data hiding scheme for digital images. In M. Zhou (Ed.), *Proceedings of the 9th International Conference on Electronic Measurement & Instruments* (p315-318). Beijing, China: IEEE Press 2009. http://dx.doi.org/10.1109/ICEMI.2009.5274054.

[17] X. Wu, D. Ou, Q. Liang, and W. Sun, A user-friendly secret image sharing scheme with reversible steganography based on cellular automata. *The Journal of Systems and Software*, 85(8), 1852-1863, 2012. http://dx.doi.org/10.1016/j.jss.2012.02.046.

[18] T.H. Chen, and C.S. Wu, Efficient multi-secret image sharing based on Boolean operations. *Signal Processing*, 91(1), 90-97, 2011. http://dx.doi.org/10.1016/j.sigpro.2010.06.012.

[19] C.C. Chen, and W.J. Wu, A secure boolean-based multi-secret image sharing scheme. *The Journal of Systems and Software*, 92(1), 107-114, 2014. http://dx.doi.org/10.1016/j.jss.2014.01.001.

[20] C.N. Yang, C.H. Chen, and S.R. Cai, Enhanced Boolean-based multi secret image sharing scheme. *The Journal of Systems and Software*, 116(1), 22-34, 2016. http://dx.doi.org/10.1016/j.jss.2015.01.031.