International Journal of Innovative Computing, Information and Control
Volume **8**, Number **1(A)**, January **2012**

ICIC International ©2012 ISSN 1349-4198
pp. 375-385

APPLY THE MODULUS FUNCTION TO SECRET IMAGE SHARING

Chi-Shiang Chan[1], Chih-Yang Lin[2] and Yu-Hsuan Lin[1]

[1]Department of Applied Informatics and Multimedia
[2]Department of Computer Science and Information Engineering
Asia University
No. 500, Lioufeng Rd., Wufeng, Taichung 41354, Taiwan
{ cschan; andrewlin }@asia.edu.tw; s98049036@msl.edu.tw

Received September 2010; revised January 2011

Abstract. *This paper proposes a method of secret image sharing by applying the Modulus Function. Initially, cover images are partitioned into four-pixel blocks. The sum value of bits that are not used to embed secret data is calculated for each four-pixel block. The range of all possible sum values is segmented to some sections in advance, and each section has its own representation value. Once the calculated sum value falls into a certain section, the representation value of this section is used to produce shadow data. Following this, the produced shadow data are embedded into a four-pixel block by using the Modulus Function. In the recovery phase, both the representation value and the shadow data are required. However, it is quite possible that the representation value has been changed owing to the modification of the sum value when embedding by using the Modulus Function. In this case, the pixels are modified to draw their sum value back to the original section. According to the experimental results, stego-images produced by the proposed method have better image quality than those produced by related works.*

Keywords: Image sharing, (t, n) threshold scheme, Modulus function

1. **Introduction.** To maintain secrecy and security of digital data traveling over the internet, data are encrypted by using one or more of several data encryption methods. Commonly, secret data are encrypted to ciphertext with a secret key and only the legal user, with the secret key, can decrypt the ciphertext back to the secret data. After that, digital signature schemes and digital multi-signature schemes [10] become popular research areas. The legal user signs a document with his/her private key (secret key) and the signed data can be authenticated by using the legal user's public key. It goes without saying that the secret key must be protected from illegal users. However, it is very likely that a careless person may lose his/her secret key. In order to prevent the secret key from being leaked by a person, a method is proposed to accomplish safe-keeping and distribution in the cryptographic system. This method partitions the secret key into shadows, which are distributed to legal users. In this way, the secret key is no longer kept by only one person.

In 1979, Shamir [5] proposed a (t, n) threshold scheme that divided a secret key into n key shadows. The n key shadows were distributed to n participants. Any t of n participants could reconstruct the secret key. If the number of participants was less than t, no information concerning the secret key was revealed through those participants. In recent years, a lot of research has been focused on improving the security and application of the (t, n) threshold scheme [2,11,14]. Since Shamir's (t, n) threshold scheme can be used to perform secret sharing, it can also be applied to image sharing. In 2002, Thien and Lin first applied a (t, n) threshold scheme to images to allow secure image sharing [6]. However, the appearance of the shadow images was sequences of meaningless codes.

This gave grabbers reasonable doubt that meaningless codes contained valuable data. To overcome this problem, data hiding [13] has been brought to this area to fool grabbers out of perceiving the existence of secret data.

In 2004, Lin and Tsai brought the concept of data hiding to image sharing [4]. Subsequently, many papers were produced concerning this topic [3,8,12]. To increase the quality of shadow images, in 2007, Yang et al. proposed their secret image sharing [12] by treating the bits, which were not used to embed shadow data, as an input of procedures to produce shadow data. Then, the produced shadow data are embedded to the least-significant bits of pixels. After that, in 2008, Chang et al. [3] proposed their secret image sharing by modifying Yang et al.'s method. Their method not only changed the bit locations used to produce shadow data but also rearranged the embedding positions of shadow data. Moreover, the authentication data were produced by using the Chinese Remainder Theorem and XOR operators. According to Chang et al.'s experimental results, the proposed method had high authentication ability and the stego-images had low degradation.

However, the above methods hid shadow data by using least-significant bit (LSB) replacement. In the area of data hiding, LSB replacement causes greatest damage to cover images [1,9], although some methods have been proposed to alleviate this degradation. Following on from this, Thien and Lin [7] proposed their hiding technique by using the Modulus Function to further improve the quality of stego-images. In this paper, the proposed method brings the Modulus Function to image sharing to further reduce the degradation of cover images. However, there is a problem when the Modulus Function is involved. The process of the Modulus Function has the possibility of modifying the unused bits of the cover pixels. Note that the shadow data are derived from those unused bits of the cover pixels. If they are modified, the secret image cannot be recovered. In order to solve this problem, the sum value of the unused bits is calculated first. The range of all possible sum values is segmented to some sections in advance, and each section has its own representation value. When a sum value falls in a certain section, the representation value of this section is involved to produce shadow data. In this way, the Modulus Function can be used to embed shadow data to cover images freely.

The rest of this paper is organized as follows. To begin with, in Section 2, we review the related works. Then, we continue to present our method in Section 3. In Section 4, we demonstrate our experimental results to show the effectiveness of our new method. Finally, the conclusions are given in Section 5.

2. **Related Work.**

In this section, two related methods are introduced. The first subsection illustrates how to use Chang et al.'s method to share a secret image. The second subsection demonstrates the image hiding scheme based on the Modulus Function.

2.1. **Chang et al.'s image sharing method.**

The purpose of image sharing is to share a secret image to n cover images to form n stego-images. Any t of n stego-images can reconstruct the secret image. No information concerning the secret image is revealed through stego-images if the number of cooperating stego-images is less than t. To achieve this goal, Chang et al.'s method first partitions each cover image into four-pixel blocks, and the secret image is also divided into blocks with t pixels in each block. Each secret block is assigned sequentially to one four-pixel block. This way, the i-th block of each cover image is related to the i-th secret block, and each secret block can be shared with its related cover blocks as follows. First of all, the $(t-1)$-degree polynomial function should be generated. Chang et al. generated their $(t-1)$-degree polynomial function as

Formula (1).

$$f(x) = a_0 + a_1x + a_2x^2 + \ldots + a_{t-1}x^{t-1} \bmod P, \tag{1}$$

where (a_0, a_1, …, a_{t-1}) are t grayscale values of a secret block and P is a prime number. In the next step, the shadow data are produced by feeding an input to the ($t-1$)-degree polynomial function. In Chang et al.'s method, the input of Formula (1) comes from the five most-significant bits of the first pixel in a four-pixel block. That is $\left(p^{(1)}_{(1)}, p^{(1)}_{(2)}, p^{(1)}_{(3)}, p^{(1)}_{(4)}, p^{(1)}_{(5)}\right)$ in Figure 1. Since the shadow data have been obtained, we go on to embed them by using Chang et al.'s method. Before going further, some notations related to four-pixel blocks should be illustrated. First of all, $I^{(i)}$ represents the i-th cover image among all n cover images. Each cover image is partitioned into four-pixel non-overlapping blocks. The notation $B^{(i)}_{(j)}$ is used to denote the j-th four-pixel block in the i-th cover image. Each four-pixel block contains four pixels, $P^{(1)}$, $P^{(2)}$, $P^{(3)}$ and $P^{(4)}$. The binary representation of the k-th pixel of a four-pixel block is $\left(p^{(k)}_{(1)}, p^{(k)}_{(2)}, p^{(k)}_{(3)}, p^{(k)}_{(4)}, p^{(k)}_{(5)}, p^{(k)}_{(6)}, p^{(k)}_{(7)}, p^{(k)}_{(8)}\right)$, where k is in the range from 1 to 4. The binary symbol $p^{(k)}_{(8)}$ is denoted as the LSB bit of the pixel $P^{(k)}$.

The binary representation of the shadow data is (s_1, s_2, …, s_8). Chang et al.'s method embeds the shadow data into two bits of each pixel in a four-pixel block. After embedding the shadow data to the cover images, the stego-images can be obtained. The symbol $\hat{I}^{(i)}$ is used to denote the i-th stego-image. Similarly, the symbol $\hat{B}^{(i)}_{(j)}$ represents the j-th four-pixel block in the i-th stego-image. To be authenticated, extra bits are needed to embed as authentication data. Chang et al's method reserved four bits to keep the authentication data. The reserved bit positions are located at A_1, A_2, A_3 and A_4 as shown in Figure 1. To produce four authentication bits, Chang et al. first set the bits used to embed four authentication bits as 0; those modified four pixels are then used to produce a large number by using the Chinese Remainder Theorem. Next, the XOR operator is applied to the large number to produce four authentication bits A_1, A_2, A_3 and A_4. The relationship between these notations is shown in Figure 1.

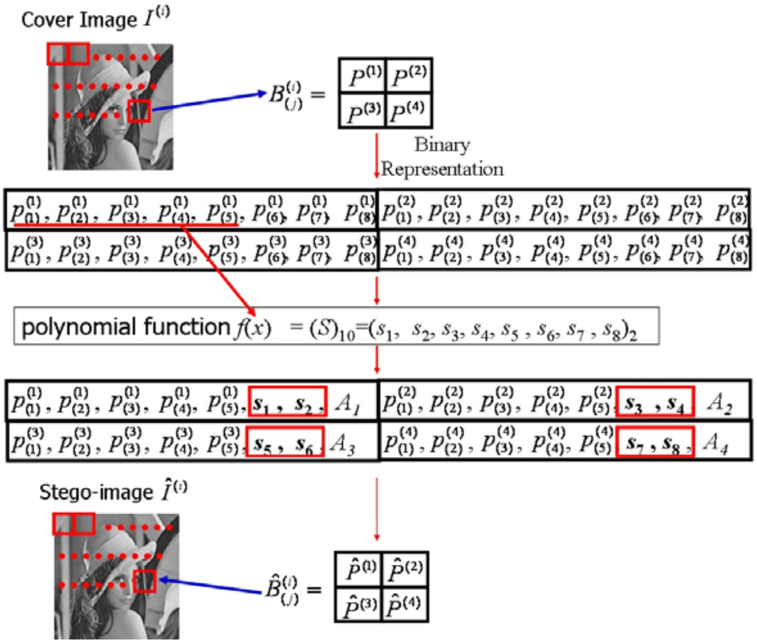

Cover Image $I^{(i)}$
$B^{(i)}_{(j)}$ =
$P^{(1)}$ $P^{(2)}$
$P^{(3)}$ $P^{(4)}$
Binary Representation
$p^{(1)}_{(1)}, p^{(1)}_{(2)}, p^{(1)}_{(3)}, p^{(1)}_{(4)}, p^{(1)}_{(5)}, p^{(1)}_{(6)}, p^{(1)}_{(7)}, p^{(1)}_{(8)}$
$p^{(2)}_{(1)}, p^{(2)}_{(2)}, p^{(2)}_{(3)}, p^{(2)}_{(4)}, p^{(2)}_{(5)}, p^{(2)}_{(6)}, p^{(2)}_{(7)}, p^{(2)}_{(8)}$
$p^{(3)}_{(1)}, p^{(3)}_{(2)}, p^{(3)}_{(3)}, p^{(3)}_{(4)}, p^{(3)}_{(5)}, p^{(3)}_{(6)}, p^{(3)}_{(7)}, p^{(3)}_{(8)}$
$p^{(4)}_{(1)}, p^{(4)}_{(2)}, p^{(4)}_{(3)}, p^{(4)}_{(4)}, p^{(4)}_{(5)}, p^{(4)}_{(6)}, p^{(4)}_{(7)}, p^{(4)}_{(8)}$
polynomial function $f(x)$ $= (S)_{10} = (s_1, s_2, s_3, s_4, s_5, s_6, s_7, s_8)_2$
$p^{(1)}_{(1)}, p^{(1)}_{(2)}, p^{(1)}_{(3)}, p^{(1)}_{(4)}, p^{(1)}_{(5)}, s_1, s_2, A_1$
$p^{(2)}_{(1)}, p^{(2)}_{(2)}, p^{(2)}_{(3)}, p^{(2)}_{(4)}, p^{(2)}_{(5)}, s_3, s_4, A_2$
$p^{(3)}_{(1)}, p^{(3)}_{(2)}, p^{(3)}_{(3)}, p^{(3)}_{(4)}, p^{(3)}_{(5)}, s_5, s_6, A_3$
$p^{(4)}_{(1)}, p^{(4)}_{(2)}, p^{(4)}_{(3)}, p^{(4)}_{(4)}, p^{(4)}_{(5)}, s_7, s_8, A_4$
Stego-image $\hat{I}^{(i)}$
$\hat{B}^{(i)}_{(j)}$ =
$\hat{P}^{(1)}$ $\hat{P}^{(2)}$
$\hat{P}^{(3)}$ $\hat{P}^{(4)}$

Figure 1. The relations between the notations

In the recovery phase, the first step extracts the shadow data and the five most-significant bits from each four-pixel block. By referring to two kinds of data extracted from the cover images, the original $(t-1)$-degree polynomial function can be rebuilt by using Largrange Interpolation. Since the polynomial function can be obtained, the coefficients of the polynomial function can be known, that is, $(a_0, a_1, \ldots, a_{t-1})$. Note that $(a_0, a_1, \ldots, a_{t-1})$ are the grayscale values of the secret pixels in a secret block.

An example is given in Figure 2 to illustrate Chang et al.'s method. There are two important points that should be noticed. First, the shadow data $(207)_{10}$ is produced from the five most-significant bits of the first pixel. Therefore, those bits cannot be modified when embedding the shadow data to this four-pixel block. Otherwise, the original $(t-1)$-degree polynomial function cannot be rebuilt, and secret pixels cannot be recovered. The second point is the manner of embedding in Chang et al.'s method. The second and the third shadow data $(000)_2$ and $(110)_2$ are embedded in two cover pixels $(127)_{10}$ and $(192)_2$ by using LSB replacement. However, LSB replacement causes the greatest damage to cover images. In the following sections, we try to improve the quality by using the Modulus Function.

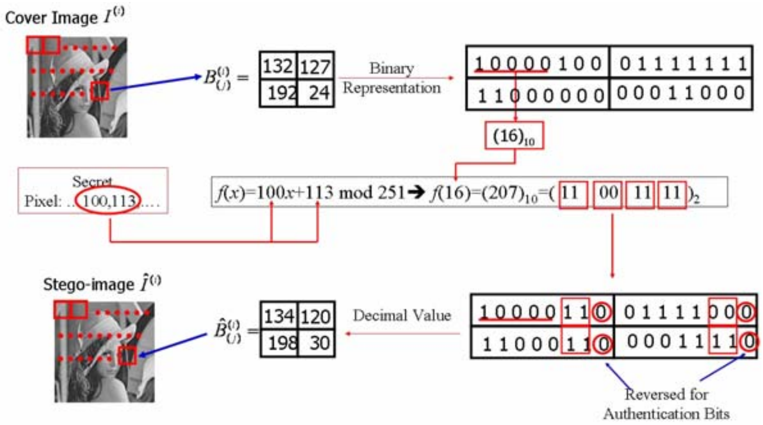


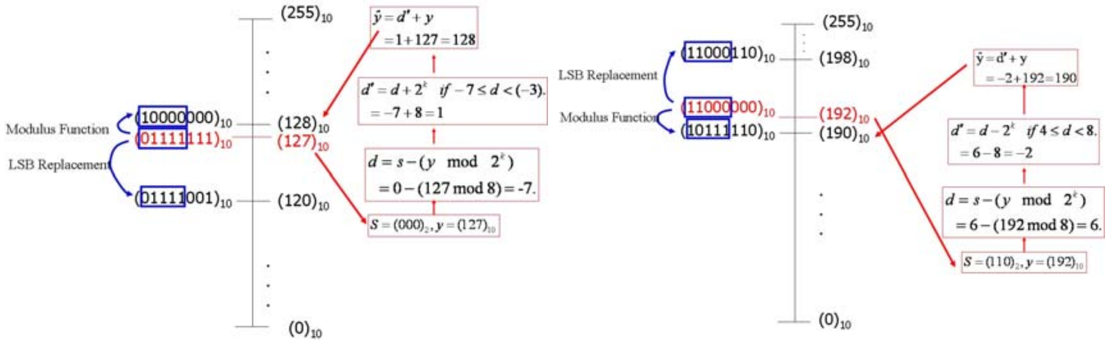
FIGURE 2. An example of Chang et al.'s method

2.2. **Thien and Lin's image hiding scheme.** In 2003, Thien and Lin [7] proposed an image hiding scheme based on the Modulus Function. To describe Thien and Lin's method, we assume that a secret value s with k-bit length is going to be embedded into a cover pixel with value y. The final purpose of Thien and Lin's method is to modify the grayscale pixel y to $\hat{y}$ such that $\hat{y} \bmod 2^k = s$. Moreover, the modified pixel $\hat{y}$ should be the closest value to the original pixel y among all possible values. In Thien and Lin's method, the difference value d is computed first by using the following equation:

$$d = s - (y \bmod 2^k). \tag{2}$$

The difference value can be further modified so that $\hat{y}$ becomes closer to y. The way to modify the difference value is shown below.

$$d' = \begin{cases} d & \text{if } \left(\left(-\left\lfloor \frac{2^k-1}{2} \right\rfloor\right) \le d \le \left(\left\lceil \frac{2^k-1}{2} \right\rceil\right)\right), \\ d + 2^k & \text{if } \left((-2^k+1) \le d < \left(-\left\lfloor \frac{2^k-1}{2} \right\rfloor\right)\right), \\ d - 2^k & \text{if } \left(\left(\left\lceil \frac{2^k-1}{2} \right\rceil\right) < d < (2^k)\right). \end{cases} \tag{3}$$



(a) (b)

Figure 3. An example of Thien and Lin's method

Finally, the modified value $\hat{y}$ can be obtained by adding the modified difference value d' to the original grayscale pixel y.

$$\hat{y} = d' + y. \tag{4}$$

An example is given below to demonstrate Thien and Lin's method. The value of k is set as 3, indicating that three secret bits are going to be embedded into a grayscale pixel. In Figure 3(a), the value of the original host pixel y is assumed to be $(127)_{10}$, and the value of 3-bit secret pixel s is $(0)_{10}$. The binary representations of the host pixel and the secret pixel are $(01111\underline{111})_2$ and $(000)_2$, respectively. If the LSB replacement is applied, the final result is $(01111\underline{000})_2$ whose decimal value is $(120)_{10}$. However, by using the Modulus Function, the value of the stego pixel $\hat{y}$ is 128, with the binary representation being $(10000\underline{000})_2$. The three least-significant bits of two modifications are equal to secret data $(000)_2$. That means secret data can be obtained from two modifications. However, $(128)_{10}$ is closer to $(127)_{10}$ comparing with $(120)_{10}$. Another example is shown in Figure 3(b).

3. **The Proposed Method.**

In this section, the Modulus Function is involved to reduce the degradation of the cover images. Before going further, let us describe the key features and important advantages compared with other methods. Although there are many methods proposed to do secret image sharing, these methods embedded the data by using the LSB replacement method. For example, Yang et al.'s method [12] and Chang et al.'s method [3] used different methods to produce sharing data and authentication data. However, both methods embedded those data by using the LSB replacement method. If the secret image sharing methods embed their data by using the LSB replacement method, it can be guaranteed that applying the proposed method on them can further reduce the degradation. This is the important advantages of the proposed method. The original way of producing authentication data in Yang et al.'s and Chang et al.'s methods can be retained. For this reason, there is no description about producing authentication data in the following paragraphs.

However, the Modulus Function cannot be applied to secret image sharing directly. The reason is that the Modulus Function may modify the unused bits which are used to be an input of Formula (1). In the example in Figure 2, the value of the first pixel is $(132)_{10}$ and its binary representation is $(10000\underline{100})_2$. The five most-significant bits of the first pixel is $(16)_{10}$. We assume that the 3-bit secret data is $(000)_2$. By applying the Modulus Function, the final result is $(136)_{10}$ and its binary representation is $(10001\underline{000})_2$. The five most-significant bits of the first pixel have been changed to $(17)_{10}$. It goes without saying that the secret pixels cannot be recovered through the modified pixel.

This paper attempts to eliminate the influence of the above problem. The proposed method partitions the bits in a pixel into two parts. One part represents the bits used to embed data while the other part is the remaining bits. The second part is named as *Unused Bits*, as shown in Figure 4. The part, *Unused Bits*, is not used for embedding purposes. There is an important point that should be described in this stage. The maximum absolute difference value of *Unused Bits* between the original pixel and the embedded pixel is 1. The reason is that Formula (3) will make the distance value d' which denotes the distance between the original pixel and the embedded pixel in the range from $-\lfloor\frac{2_k-1}{2}\rfloor$ to $\lfloor\frac{2_k-1}{2}\rfloor$. If the value of k is set as 3, then the distance value is in the range from –3 to 4. If the value of *Unused Bits* is modified more than two, the modification quantity of a pixel should be larger than 8. However, this is impossible, because once the modification quantity of a pixel is larger than 8, Formula (3) will modify it to the range from –3 to 4. Therefore, we can be sure that distance values of *Unused Bits* between the original pixel and the embedded pixel may be 1, 0 or -1. This concept drives us to sum up all *Unused Bits* of the four pixels. Therefore, the proposed method generates the input x of the $(t-1)$-degree polynomial function as follows.

$$x = \sum_{i=1}^{4} \left(P^{(i)} \gg k\right), \tag{5}$$

where k is the number of bits used to embed shadow data and authentication data for each pixel. The symbol $P^{(i)}$ is the i-th pixel in a four-pixel block as described in Section 2.1. The operator $\gg$ indicates a right shift operator. The shadow data can be produced by feeding the value calculated by Formula (5) into the $(t-1)$-degree polynomial function. The shadow data are then embedded into the four-pixel block by using the Modulus Function. The procedures are shown in Figure 4.

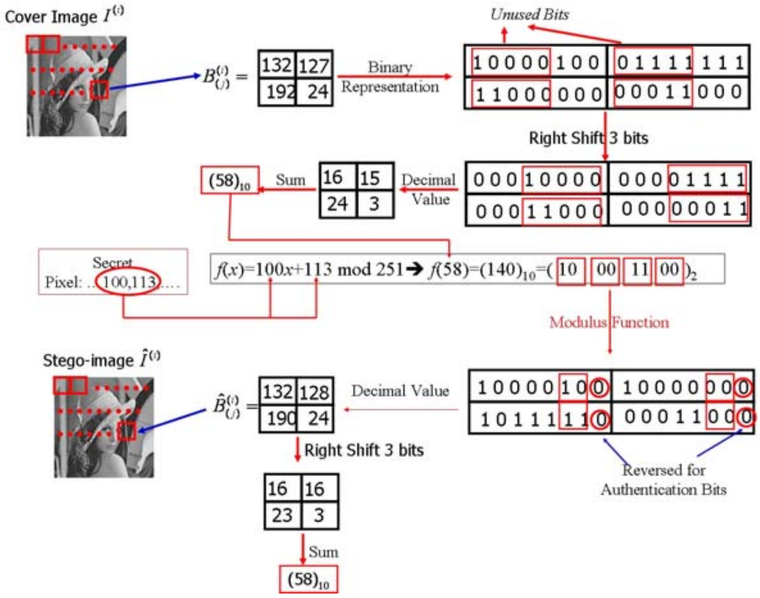


FIGURE 4. Procedures of the proposed method

The bits denoted as 0 in Figure 4 are reserved for authentication data. In the example in Figure 4, we can see that the second and the third shadow data $(000)_2$ and $(110)_2$ are embedded into two cover pixels $(127)_{10}$ and $(192)_{10}$, as the example in Figure 2. The final results are $(128)_{10}$ and $(190)_{10}$, by using the proposed method. It is easy to see that these two results are closer to the two original pixels $(127)_{10}$ and $(192)_{10}$ than the pixels produced by using a LSB replacement.

In the reconstruction phase, if the total sum of *Unused Bits* is the same as the original one, it is guaranteed that the secret data can be recovered successfully, just as in the example in Figure 5.

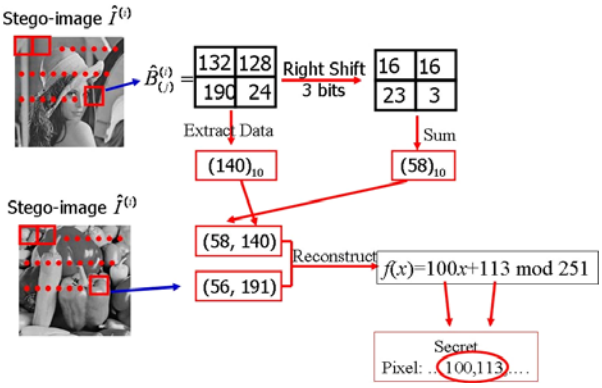


FIGURE 5. The secret reconstructing phase

However, this case may not happen all the time. The total sum value of *Unused Bits* is changed randomly after embedding shadow data by using the Modulus Function. To handle this case, the range of all possible sum values is segmented into sections in advance. One value is selected to represent each section. When a sum value falls in a section, the representation value of this section is taken as an input of the $(t-1)$-degree polynomial function to produce shadow data. In the proposed method, each section contains four values, and the representation value of each section can be calculated according to the formula below.

$$\overline{x} = x - (x \bmod 4), \tag{6}$$

where x is the sum value from Formula (5), and $\overline{x}$ is the representation value for x. Although the range of possible sum values is segmented into some sections, the sum value still has a chance to move to another section after embedding the shadow data. An example is illustrated in Figure 6. Note that $\overline{x}$, $\overline{x}+1$, $\overline{x}+2$ and $\overline{x}+3$ are in the same section, but $\overline{x}+4$ is in another section; assuming that the sum value is $\overline{x}+3$ after using Formula (5). According to Formula (6), the representation value is $\overline{x}$. Therefore, the representation value is fed into the $(t-1)$-degree polynomial function to produce the shadow data. Then, the shadow data is embedded back to the four-pixel block by using the Modulus Function.

Unfortunately, assume that the sum value of the embedded four-pixel block is $\overline{x}+4$ which belongs to another section. In the recovery phase, the secret data cannot be recovered by using the error representation value. Under this case, the additional procedures are needed to draw the sum value back to the original section. First of all, it assumes that the sum value after embedding the shadow data is $\hat{x}$. The distance x_d between $\hat{x}$ and the margin of the section is calculated from:

$$x_d = \begin{cases} 0 & \text{if } ((x \gg 2) = (\hat{x} \gg 2)), \\ \hat{x} - (\overline{x}+3) & \text{if } (((x \gg 2) = (\hat{x} \gg 2)) \text{ and } (\hat{x} > x)), \\ \hat{x} - \overline{x} & \text{if } (((x \gg 2) = (\hat{x} \gg 2)) \text{ and } (\hat{x} < x)). \end{cases} \tag{7}$$

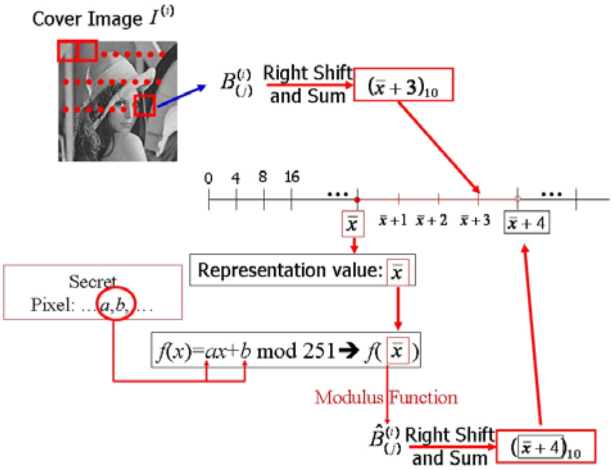


FIGURE 6. The sum value is moved to another section

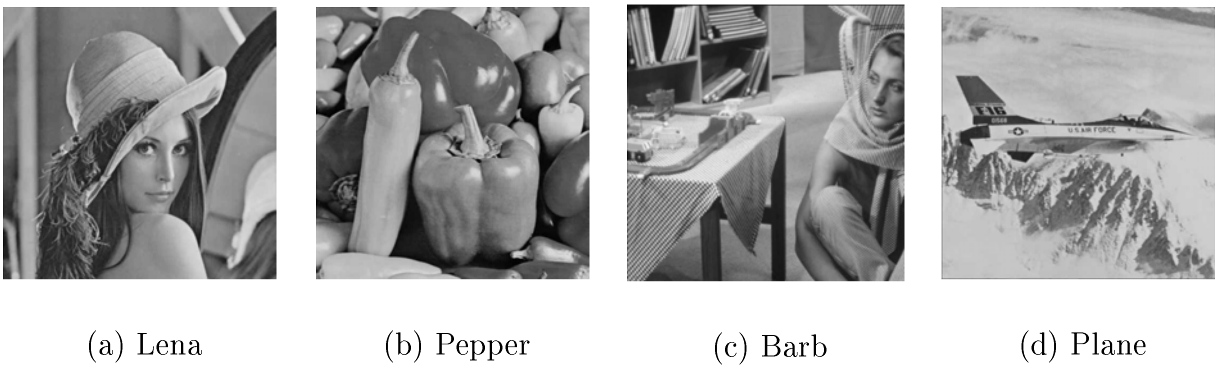

(a) Lena (b) Pepper (c) Barb (d) Plane

FIGURE 7. The test images

Once knowing the distance x_d, the sum value that comes from the embedded pixels can be adjusted according to the distance value. The adjusting procedures try to modify pixels to make the degradation as little as possible. The modified pixel $P'^{(i)}$ for each pixel in the four-pixel block can be calculated according Formula (8):

$$P'^{(i)} = \begin{cases} \left(\left(\left(\hat{P}^{(i)} \gg k\right) - 1\right) \ll k\right) + \left(\hat{P}^{(i)} \bmod 2^k\right) & \text{if } (x_d > 1), \\ \left(\left(\left(\hat{P}^{(i)} \gg k\right) + 1\right) \ll k\right) + \left(\hat{P}^{(i)} \bmod 2^k\right) & \text{if } (x_d < 1). \end{cases} \quad (8)$$

where $\hat{P}^{(i)}$ is the i-th pixel in a four-pixel block after embedding the shadow data by using the Modulus Function. For each pixel, the distance between $P'^{(i)}$ and original pixel $P^{(i)}$ is calculated. The minimal distance is the winner and its pixel value $\hat{P}^{(i)}$ will be replaced with $P'^{(i)}$. The sum value is now one step closer to the original section. The procedures above are performed x_d times, until the sum value is finally drawn back to the original section.

4. **Experimental Results.** In this section, the experimental results of the proposed method are demonstrated. For comparison, Yang et al.'s and Chang et al.'s experimental results are also shown in this section. In the experiments, the secret image was Plane shown in Figure 7(d) with size of 128 × 128 pixels. Three cover images with size of 256 × 256 pixels were shown in Figures 7(a)-7(c). They were Lena, Pepper and Barb.

Three different cover images were used to perform secret image sharing. In other words, the secret image was shared among three cover images, and with any two of three stego-images the secret image could be recovered. The way to estimate the quality of stego-images was the peak signal to noise ratio (*PSNR*), calculated from the following formula:

$$PSNR = 10 \times \log \frac{(255)^2}{MSE} dB \tag{9}$$

here, MSE means the mean square error, and is derived from the square errors of all pixels.

$$MSE = \frac{1}{w \times h} \sum_{I=1}^{w} \sum_{J=1}^{h} (\alpha(I,J) - \beta(I,J))^2 \tag{10}$$

The symbols $\alpha(I, J)$ and $\beta(I, J)$ represent the pixel values at the position (I, J) in the stego-image and the original image, respectively. The symbols w and h represent the pixel numbers for the width and the height of the image, respectively.

In the first experiment, different lengths of the authentication bits are embedded into stego-images. Because the bit lengths are different, the numbers of LSB bits reserved for embedding are different, which influences the Formula (5) and Formula (8). To adjust Formula (5) and Formula (8), the bit number that each pixel needs to right shift is related to the total bit number of embedded data. For example, if the number of authentication bits is 1, that means one bit of the first pixel in the four-pixel block should be reserved for embedding the authentication bit. Except for the authentication bit, two bits are still needed to embed the shadow data. Therefore, the first pixel must be right shifted three bits while the other pixels are right shifted two bits, as shown in Figure 8. Following from the above example, the first pixel needs to be right shifted three bits when calculating the value of Formula (8). That is, the value of k in the first pixel in Formula (8) should be three, and in the other pixels should be two. The remaining procedures are the same as described in Section 3.

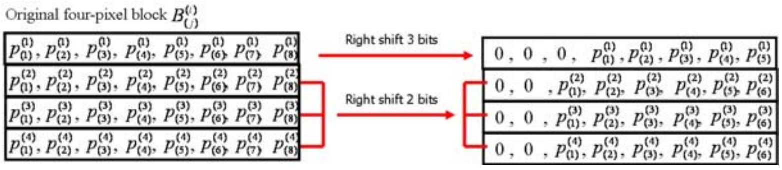


FIGURE 8. An example with one authentication bit

The *PSNR* values of the stego-images in the proposed method are shown in Table 1. It is no surprise that *PSNR* values relate to the number of authentication bits. Another point is that the image qualities of Lena, Pepper and Barb are degraded gradually. Note that input values of the $(t-1)$-degree polynomial function at the same position of different cover images cannot be the same. When dealing with the first cover image, the condition need not be considered. However, when the second cover image is processed, the input value of Formula (5) cannot be same as the value at the same position of the first cover image. If this happens, the pixels of the four-pixel block in the second image should be modified. It goes without saying that the third image has more chances of being modified. That is the reason why the image qualities of Lena, Pepper and Barb are degraded gradually. Although the third image is degraded the most, the image quality is still acceptable.

In the second experiment, the comparisons between the proposed method and the related methods are demonstrated. There are two related methods, Yang et al.'s method

Table 1. The *PSNR* values of the stego-images

Number of bits	Stego-Images		
	Lenna	Pepper	Barb
One authentication bit	46.44	46.24	45.99
Two authentication bits	44.82	44.61	44.39
Three authentication bits	43.63	43.34	42.89
Four authentication bits	42.60	42.19	41.48

and Chang et al.'s method. Yang et al.'s method only embedded one authentication bit in each four-pixel block while Chang et al.'s method embedded four authentication bits in each four-pixel block. Therefore, the proposed method is adjusted to embed one authentication bit and four authentication bits, respectively. The *PSNR* values of the stego-images using different methods are shown in Table 2 and Table 3. As shown in Table 2 and Table 3, the stego-images produced by the proposed method have a better quality than those produced by other methods. Owing to the different bit lengths of authentication data, the qualities of stego-images produced by using Yang et al.'s method are higher than those produced by using Chang et al.'s method. However, the *PSNR* values of stego-images produced by using the proposed method are always higher than those produced by using other methods.

Table 2. The *PSNR* values of the stego-images with one authentication bit

Methods	Stego-Images		
	Lenna	Pepper	Barb
The proposed method	46.44	46.24	45.99
Yang et al.'s method	44.70	44.49	44.48

Table 3. The *PSNR* values of the stego-images with four authentication bits

Methods	Stego-Images		
	Lenna	Pepper	Barb
The proposed method	42.60	42.19	41.48
Chang et al.'s method	41.01	40.77	40.52

5. Conclusions.

Shamir's (t, n) threshold scheme was used to achieve secret sharing in the area of cryptographic systems. In 2002, Thien and Lin applied the (t, n) threshold scheme to images to achieve image sharing. Subsequently, image sharing became the focus of many researchers. In this paper, the Modulus Function is involved to reduce the degradation of cover images, and the proposed method can adjust bit lengths of authentication data to fit users' requests. According to experimental results, the proposed method is superior to Chang et al.'s methods and Yang et al.'s methods. All in all, this paper proposes a secret image sharing method that can produce high quality stego-images.

Acknowledgment. This work is partially supported by National Science Council, Taiwan under contract number NSC 99-2221-E-468-013. The authors also gratefully acknowledge the helpful comments and suggestions of the reviewers, which have improved the presentation.

REFERENCES

[1] C. C. Chang, J. Y. Hsiao and C. S. Chan, Finding optimal LSB substitution in image hiding by dynamic programming strategy, *Pattern Recognition*, vol.36, no.7, pp.1583-1595, 2003.

[2] D. Catalano and R. Gennaro, New efficient and secure protocols for verifiable signature sharing and other applications, *Journal of Computer and System Sciences*, vol.61, no.1, pp.51-80, 2000.

[3] C. C. Chang, Y. P. Hsieh and C. H. Lin, Sharing secrets in stego images with authentication, *Pattern Recognition*, vol.41, pp.3130-3137, 2008.

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

[5] A. Shamir, How to share a secret, *Communication of the ACM*, vol.22, no.11, pp.612-613, 1979.

[6] C. C. Thien and J. C. Lin, Secret image sharing, *Computer and Graphics*, vol.26, pp.765-770, 2002.

[7] C. C. Thien and J. C. Lin, A simple and high-hiding capacity method for hiding digit-by-digit data in images based on modulus function, *Pattern Recognition*, vol.36, no.12, pp.2875-2881, 2003.

[8] M.-H. Tsai, Y.-B. Lin and C.-M. Wang, Image sharing with steganography and cheater identification, *International Journal of Innovative Computing, Information and Control*, vol.6, no.3(A), pp.1165-1178, 2010.

[9] R. Z. Wang, C. F. Lin and J. C. Lin, Hiding by optimal LSB substitution and genetic algorithm, *Pattern Recognition*, vol.34, no.3, pp.671-683, 2001.

[10] L. J. Wang and J. J. R. Chen, Novel digital multisignature scheme, *ICIC Express Letters*, vol.4, no.4, pp.1251-1256, 2010.

[11] C. H. Wang and T. Hwang, (t, m) threshold and generalized ID-based conference key distribution system, *Applied Mathematics and Computation*, vol.112, no.2-3, pp.181-191, 2000.

[12] 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.

[13] Z. Yin, C. Chang and Y. Zang, An information hiding scheme based on (7,4) hamming code oriented wet paper codes, *International Journal of Innovative Computing, Information and Control*, vol.6, no.7, pp.3121-3130, 2010.

[14] Z. J. Zhang, Robust multiparty quantum secret key sharing over two collective-noise channels, *Physical A: Statistical Mechanics and Its Applications*, vol.361, no.1, pp.233-238, 2006.