Journal of Information Hiding and Multimedia Signal Processing ©2012 ISSN 2073-4212
Ubiquitous International Volume **3**, Number **4**, October **2012**

A User-Friendly Image Sharing Scheme Using JPEG-LS Median Edge Predictor

Chi-Shiang Chan

Department of Applied Informatics and Multimedia
Asia University
500, Lioufeng Rd., Wufeng, Taichung 41354, Taiwan
CSChan@asia.edu.tw

Chin-Chen Chang

Department of Information Engineering and Computer Science
Feng Chia University
100, Wenhwa Rd., Seatwen, Taichung 40724, Taiwan
Department of Biomedical Imaging and Radiological Science
China Medical University
91, Hsueh-Shih Road, Taichung,40402 Taiwan
Department of Computer Science and Information Engineering
Asia University
500, Lioufeng Rd., Wufeng, Taichung 41354, Taiwan
alan3c@gmail.com

Hung P. Vo

Department of Information Engineering and Computer Science
Feng Chia University
100, Wenhwa Rd., Seatwen, Taichung 40724, Taiwan
vophuochung@gmail.com

Received July 2012; revised September 2012

Abstract. *Developed by Yang et al. in 2007, a user-friendly (k, n)-threshold scheme based on Shamir's polynomial with different primes calculates how pixels in a block and their left pixels differ from each other. Based on those differences, the prime number for Shamir's polynomial can be determined as well as the differences distributed to shares by using Shamir's polynomial. This work attempts to calculate the differences between pixels in a block by using the JPEG-LS median edge predictor. Given the role of this predictor, the generated differences refer to both the left pixel and its neighboring pixels, thus diminishing the values of differences and enhancing the reconstructed image quality. Experimental results demonstrated the effectiveness of using the JPEG-LS median edge predictor.*
Keywords: image sharing, secret sharing, secret image sharing, user-friendly shadow image

1. **Introduction.** Recent advances in computer networks have popularized the transmission of digital media data, as evidenced by lots of condential images transferred via Internet, despite the inability to ensure condentiality via such communications. Therefore, despite the increasing attention to the security of condential images, a secret image still cannot be recovered if lost or destroyed during transmission. Overcoming this problem involves sharing a secret image for n participants by using the so-called (k, n)-threshold

secret sharing schemes. Shamir [5] and Blakley [1] pioneered the concept of secret sharing in 1979, independently. Secret data is shared into n shadows and the original data can be restored from any k or more shadows. Thien and Lin (2002) developed a (k, n)-threshold secret image sharing scheme [6] based on Shamir's scheme. The scheme considers pixels in a secret image as coecients of the (k-1)-degree polynomial function to share the secret image into n shadow images. The shadow images are distributed directly to intended participants separately. Only the collaboration of any k or more of n authorized participants can recover the secret image. Moreover, the size of each shadow image generated by the (k, n)-threshold scheme is normally smaller than that of the secret image.

The practical application of a (k, n)-threshold secret image sharing scheme is that a secret image can be distributed to storage branches securely. Moreover, the scheme can also achieve fault tolerance and fast transmission [3, 6]. More precisely, the shadow images produced from a secret image is stored in storage branches separately. When authorized user needs the original secret image, he/she can extract any k or more of n shadow images from storage branches simultaneously in parallel. Furthermore, the size of each shadow image is smaller than that of the original secret image. Owing to the descriptions above, the scheme can achieve fast transmission. In the aspect of fault tolerance, only the collaboration of any k or more of n authorized participants can recover the secret image. If any of storage branches crashes, the original secret image still can be recovered by extracting another shadow image from other storage branches.

However, Thien and Lin's scheme normally generates meaningless shadow images which are not easy to manage in the local storage disks. The manager can not identify shadow images from each other because they are noise-like shadow images. Thien and Lin [7] then devised an image-sharing scheme with user-friendly shadow images in 2003, in which shadow images resembling a shrunken replica of the original image are generated. Yang *et al.* [01] developed their scheme in 2007, in which different polynomials with different primes are used for sharing blocks. According to the experimental results, the recovered image quality of Yang *et al.*'s scheme is better than that of Thien and Lin's scheme.

However, the image quality of these two user-friendly image-sharing schemes [7, 10] is still inferior, making them infeasible for medical, military, or artistic applications [8, 9]. As we know that digital versions of some special images such as medical images, military images, or artistic applications just allow a slight amount of modifications. A large amount of modications may destroy the meaning of the contents in those materials. Since the contents in those images are important, they are usually compressed by using lossless image compression such as JPEG lossless image compression, and the compressed image are transmitted via Internet. It is guaranteed that the compressed image can be recovered to its original version without any loss through JPEG lossless image compression. Additionally, the uncompressed image formant such as bitmap file format (BMP) is usually used in those images.

This work presents a user-friendly sharing scheme, in which different primes and JPEG-LS median edge predictor are used for sharing blocks. Owing to the property of JPEG-LS median edge predictor, the pixel value difference between the predicted pixel and the secret pixel becomes small. The frequency of using small prime numbers increases. Therefore, the qualities of both the reconstructed secret image and shadow images in the proposed scheme are better than those in Yang *et al.*'s scheme.

The rest of this paper is organized as follows. Section 2 briefly reviews pertinent literature. Section 3 then describes the proposed scheme in detail. Section 4 summarizes the experimental results. Conclusions are finally drawn in Section 5.

2. **Related Works.** In this Section, Yang *et al.*'s user-friendly image sharing scheme is illustrated first in Subsection 2.1. Then, the concept of JPEG-LS median edge predictor (MED) which will be used in the proposed scheme is described in Subsection 2.2.

2.1. **Yang *et al.*'s image sharing scheme.** Yang *et al.*'s image sharing scheme contains two steps. The final purpose of the first step is to record the prime number used in the current block to its previous block. In a basic (2, n) scheme, Yang *et al.* initiate the sharing by selecting a set of four primes $\{p_0, p_1, p_2, p_3\}$ forced by $p_0 < p_1 < p_2 < p_3 \leq 251$. The indicators of $\{p_0, p_1, p_2, p_3\}$ are $\{(0, 0), (0, 1), (1, 0), (1, 1)\}$, respectively. The secret image is then divided into two-pixel non-overlapping blocks. The least signicant bits (LSB) of two pixels in a previous block are modified as (0, 0), (0, 1), (1, 0) or (1, 1) to indicate the prime number p_0, p_1, p_2 or p_3 used in the current block. Assume that (P_{c-2}, P_{c-1}) and (P_c, P_{c+1}) are the previous block and the current blocks, respectively. The least signicant bits of two pixels in the previous block are modied as follows:

$$\begin{cases} LSB(P_{c-2}) = 0, \ LSB(P_{c-1}) = 0 \ \ for \ \ |P_{\max} - P_{c-1}| \leq (p_0 - 1)/2, \\ LSB(P_{c-2}) = 0, \ LSB(P_{c-1}) = 1 \ \ for \ \ (p_0 - 1)/2 < |P_{\max} - P_{c-1}| \leq (p_1 - 1)/2, \\ LSB(P_{c-2}) = 1, \ LSB(P_{c-1}) = 0 \ \ for \ \ (p_1 - 1)/2 < |P_{\max} - P_{c-1}| \leq (p_2 - 1)/2, \\ LSB(P_{c-2}) = 1, \ LSB(P_{c-1}) = 1 \ \ for \ \ (p_2 - 1)/2 < |P_{\max} - P_{c-1}| \leq 250 \end{cases} \tag{1}$$

where $P_{\max}$ is the pixel belonging to the current block ($P_{\max} \in \{P_c, P_{c+1}\}$) that diRers the most from P_{c-1}. Notably, $P_{\max}$ is evaluated as follows:

$$P_{\max} = \begin{cases} P_c, & if \ \ |P_c - P_{c-1}| > |P_{c+1} - P_{c-1}|, \\ P_{c+1}, & otherwise. \end{cases} \tag{2}$$

After embedding the indicator of the prime number of each current block to its previous block, the modified secret image can be obtained at the end of the rst step. We give an example in Fig. 1 to illustrate the first step of Yang *et al.*'s scheme. The prime numbers $\{p_0, p_1, p_2, p_3\}$ used in Fig. 1 are {17, 61, 131, 251}

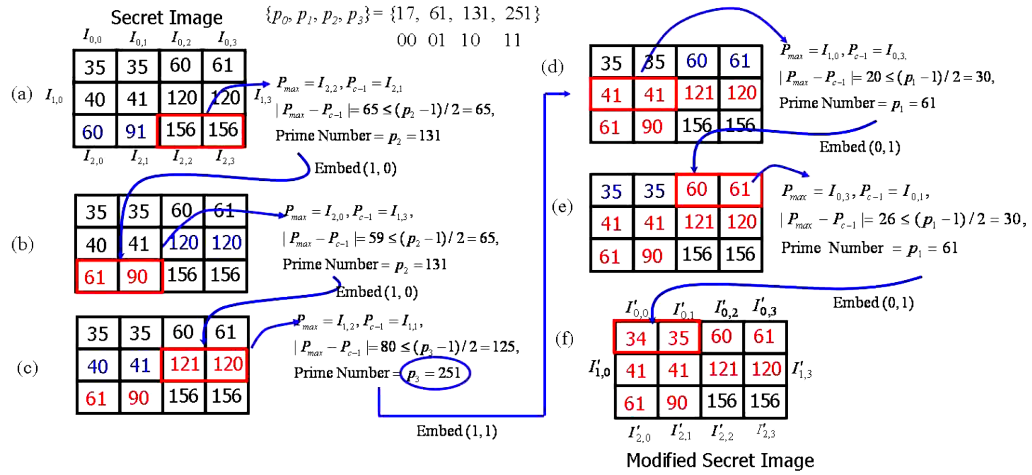


FIGURE 1. The way to produce the modified secret image in Yang *et al.*'s scheme

In Fig. 1 (a), the current block and previous block are $(I_{2,2}, I_{2,3}) = (156, 156)$ and $(I_{2,0}, I_{2,1}) = (60, 91)$, respectively. According to Formula (1), the largest pixel value difference is $|P_{\max} - P_{c-1}| = |156 - 91| = 65$, which is smaller than or equal to $(p_2 - 1)/2$. Therefore, the prime number used in the current block is p_2 whose value is 131. The indicator of the prime number p_2 is embedded to the previous block ($I_{2,0}$, $I_{2,1}$). The nal result of the

previous block becomes (61, 90). The same procedures are performed on each two-pixel block sequentially to get the nal resultant image as shown in Fig. 1(f).

The second step is the sharing step. In this step, two variables f_0 and f_1 are derived from pixels in the current block firstly. The way to derive these two variables f_0 and f_1 are evaluated as follows:

$$\begin{cases} f_0 = (P_c - P_{c-1}) + (p_j - 1)/2, \\ f_1 = (P_{c+1} - P_{c-1}) + (p_j - 1)/2, \end{cases} \quad for\ \ j \in 0, 1, 2. \tag{3}$$

$$\begin{cases} f_0 = \lfloor (P_c - P_{c-1})/2 \rfloor + (p_j - 1)/2, \\ f_1 = \lfloor (P_{c+1} - P_{c-1})/2 \rfloor + (p_j - 1)/2, \end{cases} \quad for\ \ j \in 3. \tag{4}$$

where the prime number p_j is determined by Formula (1).

The polynomial function in Yang *et al.*'s scheme can be built by taking two variables f_0 and f_1 as coecients of the polynomial function as follows:

$$S(x) = (f_0 + x \times f_1) mod\ p_j, \quad for\ \ j \in 0, 1, 2, 3. \tag{5}$$

Let $\hat{P}_i$ be the shadow pixel of the i-th shadow image and x_i be a random and unique number assigned to the i-th shadow image. The variable x_i can be treated as the identification number of the i-th shadow image. Then, shadow pixels $\hat{P}_i$ can be obtained by locating its value as close to the average pixel value in the current block as possible, but its value also satises the constraint $\hat{P}_i\ mod\ p_j = S(x_i)$.

Following the same example above in Fig. 1, assume that we have processed some blocks. Now, the current block is $(I'_{0,1}, I'_{1,1})$, and three shadow pixels will be derived from the current block. Because the least-signicant bits of two pixels in the previous block is (0, 1), the prime number used in the current block is p_1, whose value is 61. Referring to Formula (3), two coecients f_0 and f_1 are 10 and 10, respectively. Therefore, the polynomial function can be built as $S(x) = (10 + 10x)\ mod\ 61$. Assume the identification numbers of x_1, x_2, and x_3 are 1, 2, and 3, respectively. Then, the resultant values of $S(x_1)$, $S(x_2)$, and $S(x_3)$ are 20, 30, and 40, respectively. Moreover, the average value of two pixels in the current block is 41. According to the information above, the shadow pixel in the first shadow image should be 20, because 20 is closest to average value 41 but 20 also satisfies the constraint $20\ mod\ 61 = 20$. The other two shadow pixels of two shadow images can be obtained by using the same procedures.

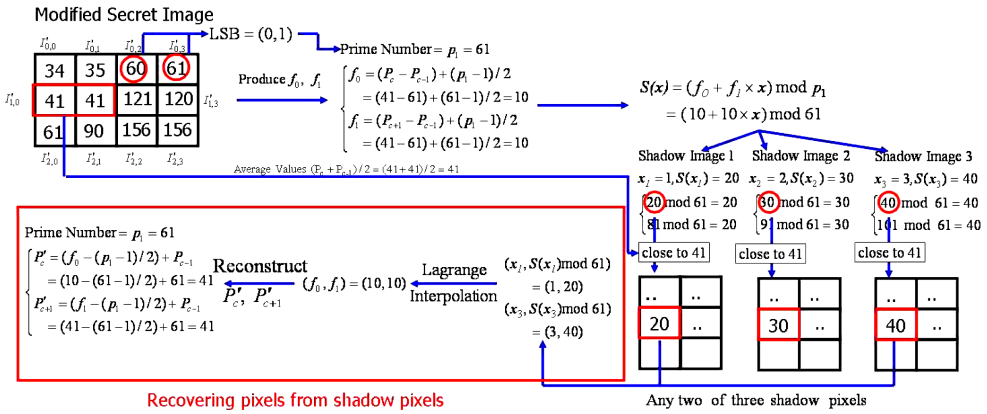


FIGURE 2. The way to produce shadow pixels

In the recovering phase, once any two or more shadow images are obtained, the pixels in the modified secret image can be restored. Assume the identication numbers of two shadow images are y_1 and y_2, respectively. Moreover, $S(y_1)$ and $S(y_2)$ can be derived by

calculating $(\hat{P}_{y_1} \ mod \ p_1)$ and $(\hat{P}_{y_1} \ mod \ p_1)$, where $\hat{P}_{y_1}$ and $\hat{P}_{y_2}$ represent two shadow pixels in two shadow images y_1 and y_2. The polynomial function can be reconstructed by using Lagrange Interpolation below [5]:

$$S(x) = (S(y_1) \times \frac{(x - y_2)}{(y_1 - y_2)} + S\,(y_2) \times \frac{(x - y_1)}{(y_2 - y_1)}) mod \ p_j \tag{6}$$

Two variables f_0' annd f_1' can be extracted from the coecients of the reconstructed polynomial function. After that, the pixels in the modied secret image can be reconstructed from f_0' and f_1'. In Fig. 2, the example uses shadow images 1 and 3 to perform the recovering procedures. According to variables $x_1 = 1$, $x_3 = 3$, $S(x_1) = 20$, and $S(x_3) = 40$, two variables f_0' annd f_1' can be extracted from the coecients of the reconstructed polynomial function which is derived by using Lagrange Interpolation, $S(x) = \left(S\,(x_1) \times \frac{(x-x_3)}{(x_1-x_3)} + S\,(x_3) \times \frac{(x-x_1)}{(x_3-x_1)}\right) \ mod \ 61 = \left(20 \times \frac{(x-3)}{(1-3)} + 40 \times \frac{(x-1)}{(3-1)}\right) \ mod \ 61 = 10x + 10 \ mod \ 61$.

Therefore, the values f_0' and f_1' are both 10. If the prime number is p_0, p_1 or p_2, two secret pixels P_c' and P_{c+1}' can be recovered through f_0' and f_1' by using the formula below:

$$\begin{cases} P_c' = (f_0' - (p_j - 1)/2) + P_{c-1}', \\ P_{c+1}' = (f_1' - (p_j - 1)/2) + P_{c-1}', \end{cases} \quad for j \in 0, 1, 2. \tag{7}$$

However, it may cause a serious problem when using p_3 as the prime number. The problem comes from using Formula (4) to calculate two coecients f_0 and f_1. An example is given in Fig. 3 to illustrate this problem. Because the prime number used in the current block is p_3, two coecients f_0 and f_1 are derived from Formula (4). It can be seen that the pixel value difference between P_{c-1} and P_{c+1} is odd. When f_1 is derived from $\lfloor (P_{c+1} - P_{c-1})/2 \rfloor + (p_3 - 1)/2$, the least-signicant bit of the pixel value difference between P_{c-1} and P_{c+1} is truncated. Therefore, the reconstructed value P_{c+1}' is not the same as its original value Pc+1 as shown in Fig. 3. Note that the least-significant bits of P_c and P_{c+1} are used to indicate the prime number for a certain block. Therefore, the least-significant bits of P_c and P_{c+1} should be recovered exactly, or it will affect the correction of the remaining recovering procedures. To overcome this problem, Yang *et al.* simply embed the least-significant bit of the pixel value difference to the second least-significant bit (i.e. 7^{th} bit) of the secret pixels in the previous block. However, this kind of modification may cause image degradation. It goes without saying that the frequency of using Formula (4) will affect the quality of the reconstructed secret image.

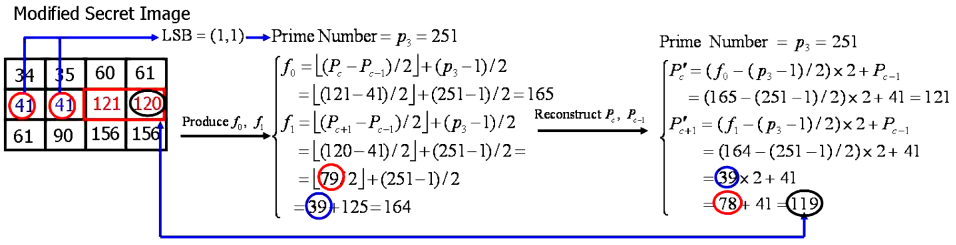


FIGURE 3. Truncating the least-significant bit of the pixel value difference

2.2. **JPEG-LS median edge predictor(MED)..** JPEG-LS [2, 4] represents the latest JPEG standard for lossless and near lossless image compression, as created in the 1990s and designed by Hewlett Packard. In JPEG-LS, edge detection of horizontal or vertical edges is estimated by examining the neighboring pixels of the current pixel X, as illustrated in Fig. 4.

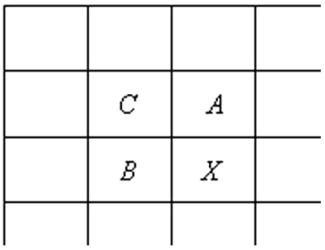


FIGURE 4. Three neighboring samples around the predicted X

The predicted value X can be evaluated by feeding C, A, and B as the parameters to the formula as follows:

$$MED\,(C,A,B) = \begin{cases} min\,(A,B)\,, & if\ \ C \geq \max(A,B) \\ man(A,B), & if\ \ C \leq mix(A,B) \\ A+B-C, & otherwise \end{cases} \tag{8}$$

An edge is detected through three pixels A, B, and C, when $C \geq \max(A,B)$ or $C \leq mix(A,B)$ as shown in Fig. 5.

3. **Proposed scheme.** As described in Subsection 2.1, the frequency of using Formula (4) will affect the quality of the reconstructed secret image. In this paper, the proposed scheme modifies Formula (4) such that the 7^{th} bit of the secret pixel needs not to be modified. Moreover, the proposed scheme uses JPEG-LS median edge predictor to predict pixel values so that the frequency of using smaller prime numbers increases. The qualities of both reconstructed secret image and shadow images are improved. In Subsection 3.1, the way to determinate the prime number for each two-pixel block is described. Then, the sharing procedures are illustrated in Subsection 3.2.

3.1. **Determining the prime number.** Determining the prime number for each block is very important. In some case, side effect may occur. Referring to the case in Fig. 1 (a), the original pixel values of $I_{2,0}$ and $I_{2,1}$ are 60 and 91, respectively. The pixel value difference between 156 and 91 is 65. According to Formula (1), the current block (156, 156) will be encoded by using the prime number p_2, and its indicator (1, 0) is recorded in the least-significant bits of the pixels in the previous block (60, 91). Therefore, the final result of the previous block becomes (61, 90) as shown in Fig. 1(b). However, it is easy to calculate that the pixel value difference between 90 and 156 has already become 66 which is larger than $(p_2 - 1)/2 = 65$. According to Formula (1), the prime number for this pixel value difference should be p_3, which conflicts with the previous decision of using p_2 as its prime number.

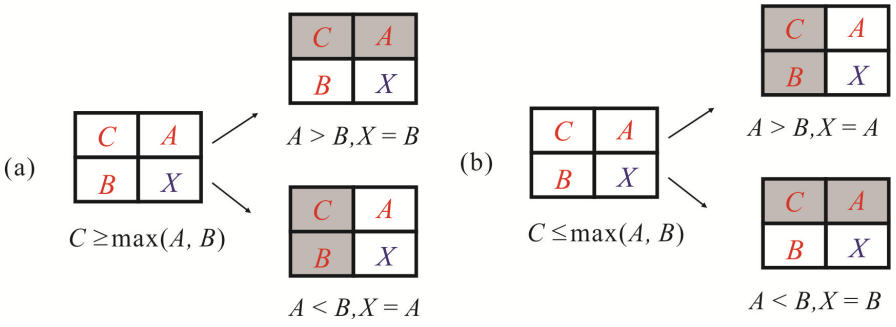


FIGURE 5. Edge detection flowcharts (a) $c \geq \max(A,B)$; (b) $c \leq \min(A,B)$

Yang *et al.*'s scheme simply replaces p_3 as its prime number and embeds the new indicator of the prime number p_3 to the previous block. The embedding order of Yang *et al.*'s scheme is from bottom to top and right to left. The predicted pixel value is only related to the right pixel of the previous block. Therefore, replacing larger prime number dose not cause any problem. However, the proposed scheme uses JPEG-LS median edge predictor to predict pixel value. The predicted pixel value is related to its neighboring pixels. Modifying any pixel will affect the predicted values of the pixels surrounding that pixel. Therefore, it is not suitable to use the embedding order of Yang *et al.*'s scheme in the proposed scheme. Therefore, the embedding order of the proposed scheme is set from top to bottom and left to right.

When determining the prime number, all possible cases are checked first to find the possible largest pixel value difference for the next block. According to the largest pixel value difference, the prime number for the next block can be determined and recorded in the current block. More precisely, assume the pixels in the current block and the next block are $(I_{i,j-1}, I_{i,j})$ and $(I_{i,j+1}, I_{i,j+2})$, respectively. First of all, the formula of truncating the least-significant bit of a pixel is illustrated as below:

$$TrunLSB(x) = x - LSB(x) \tag{9}$$

where x is a pixel value that want to truncate its least-significant bit.

Because the prime number used in the next block will be recorded in the least-significant bits of the pixels in the current block, all possible final resultant values of $I_{i,j}$ would be $TrunLSB(I_{i,j})$ or $TrunLSB(I_{i,j}) + 1$. Moreover, the possible final resultant values of $I_{i,j+1}$ and $I_{i,j+2}$ in the next block would be $TrunLSB(I_{i,j+1})$ or $TrunLSB(I_{i,j+1}) + 1$ and $TrunLSB(I_{i,j+2})$ or $TrunLSB(I_{i,j+2}) + 1$ respectively. Therefore, eight possible combinations of $(I_{i,j}, I_{i,j+1}, I_{i,j+2})$ are $(TrunLSB(I_{i,j}), TrunLSB(I_{i,j+1}), TrunLSB(I_{i,j+2}))$, $(TrunLSB(I_{i,j}), TrunLSB(I_{i,j+1}), TrunLSB(I_{i,j+2}) + 1)$,..., and $(TrunLSB(I_{i,j}) + 1, TrunLSB(I_{i,j+1}) + 1, TrunLSB(I_{i,j+2}) + 1)$. The way to calculate pixel value diRerence for one of eight possible resultant values in the next block is shown below:

$$d_{max}(k,l,m) = max \left\{ \begin{array}{l} |(TrunLSB\,(I_{i,j+1}) + l) - MED\,(I_{i-1,j}, I_{i-1,j+1}, TrunLSB\,(I_{i,j}) + k)| \\ |(TrunLSB\,(I_{i,j+2}) + m) - MED\,(I_{i-1,j+1}, I_{i-1,j+2}, TrunLSB\,(I_{i,j+1}) + l)| \end{array} \right\}. \tag{10}$$

where k, l, m belong to 0 or 1. The largest pixel value difference exists in one of eight possible resultant values in the next block as shown below:

$$d_{max} = max \left\{ \begin{array}{l} d_{max}(0,0,0), d_{max}(0,0,1), d_{max}(0,1,0), d_{max}(0,1,1) \\ d_{max}(1,0,0), d_{max}(1,0,1), d_{max}(1,1,0), d_{max}(1,1,1) \end{array} \right\} \tag{11}$$

An example is given in Fig. 6. The prime number used in the next block (120, 120) is recorded in the current block (40, 41). Because the proposed scheme only modies the least-significant bit of pixels, the possible resultant values of $I_{1,1}$, $I_{1,2}$ and $I_{1,3}$ are (40, 120, 120), (40, 120, 121), (40, 121, 120), (40, 121, 121) , (41, 120, 120), (41, 120, 121), (41, 121, 120) and (41, 121, 121). Then, the largest pixel value difference can be calculated by using Formula (10) and (11), and its value is 61. According to the value of the largest pixel value difference, the prime number used in the next block is p2, and its indicator (1, 0) is going to record to the current block (40, 41). Finally, the resultant current block becomes (41, 40).

3.2. **Sharing and recovering phase.** There are two steps in the proposed scheme. In a basic (2, n) scheme, the first step initiates the sharing by selecting a set of four primes $\{p_0, p_1, p_2, p_3\}$ as described in Subsection 2.1. Assume the pixels in the current block and next block are $(I_{i,j-1}, I_{i,j})$ and $(I_{i,j+1}, I_{i,j+2})$, respectively. Obeying the same procedures

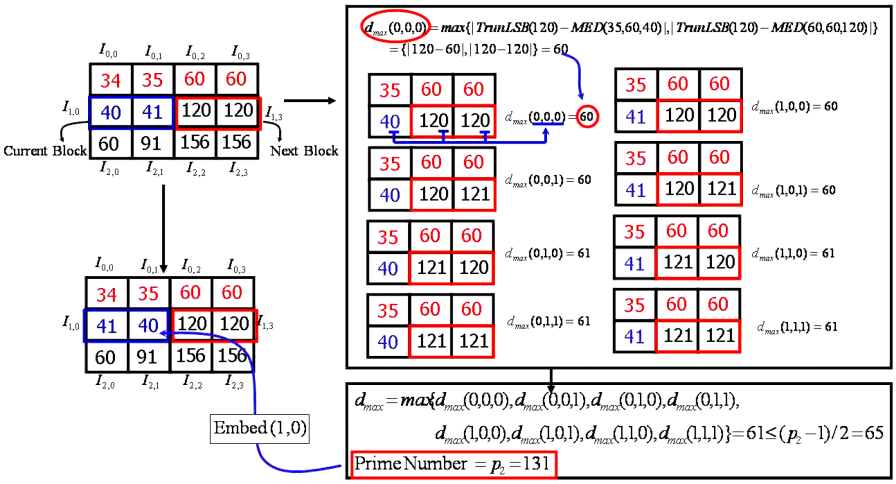


FIGURE 6. Determining the prime number

in Section 3.1, the largest pixel value difference d_{max} for the next block can be calculated. And, the prime number for the next block can also be determined according to the value of the largest pixel value difference d_{max} as follows:

$$\begin{cases} LSB\left(I_{i,j-1}\right)=0, LSB(I_{i,j})=0 \ \ for \ \ d_{max} \leq (p_0-1)/2 \\ LSB\left(I_{i,j-1}\right)=0, LSB(I_{i,j})=1 \ \ for \ \ (p_0-1)/2 < d_{max} \leq (p_1-1)/2 \\ LSB\left(I_{i,j-1}\right)=1, LSB(I_{i,j})=0 \ \ for \ \ (p_1-1)/2 < d_{max} \leq (p_2-1)/2 \\ LSB\left(I_{i,j-1}\right)=1, LSB(I_{i,j})=1 \ \ for \ \ (p_2-1)/2 < d_{max} \leq 250 \end{cases} \tag{12}$$

Since the prime number used in the next block is known, the indicator of the prime number can be embedded in the current block. After embedding the indicators of the prime numbers used in the next blocks to their current blocks, the modified secret image can be obtained at the end of the first step. We give an example in Fig. 7 to illustrate the first step of the proposed scheme.

Note that the pixel values which are out of the image boundary are set as 0. For example, the pixel value of $I_{-1,1}$ is 0. In Fig. 7(a), the largest possible pixel value difference occurs when the values of $I_{0,1}$ and $I_{0,2}$ are 34 and 61,respectively. Under this situation, the predicted value of $I_{0,2}$ is $MED\left(I_{-1,1}, I_{-1,2}, I_{0,1}\right) = MED\left(0,0,34\right) = 34$ Therefore, the largest pixel value difference d_{max} is $|61-34|=27$. According to Formula (12), the prime number used in the next block (60, 61) is p_1, and the indicator (0, 1) is embedded in the least-significant bits of the pixels in the current block (35, 35). Finally, the result of the current block becomes (34, 35). The same procedures are performed to produce the modified secret image as shown in Fig. 7(f).

The second step is the sharing step. In this step, two variables f_0 and f_1 are derived from pixels in the modied secret image firstly. The way to derive two variables f_0 and f_1 are evaluated as follows:

$$\begin{cases} f_0 = \left(I'_{i,j} - MED\left(I'_{i-1,j-1}, I'_{i-1,j}, I'_{i,j-1}\right)\right) + (p_j-1)/2 \\ f_1 = \left(I'_{i,j+1} - MED\left(I'_{i-1,j}, I'_{i-1,j+1}, I'_{i,j}\right)\right) + (p_j-1)/2 \end{cases} for j \in 0,1,2. \tag{13}$$

$$\begin{cases} f_0 = I'_{i,j} \\ f_1 = I'_{i,j+1} \end{cases} for \ \ j \in 3. \tag{14}$$

where the prime number p_j is determined by Formula (12), and $(I'_{i,j},\ I'_{i,j+1})$ is the current block of the modified secret image. The rest parts of sharing procedures are the same as Yang *et al.*'s scheme.

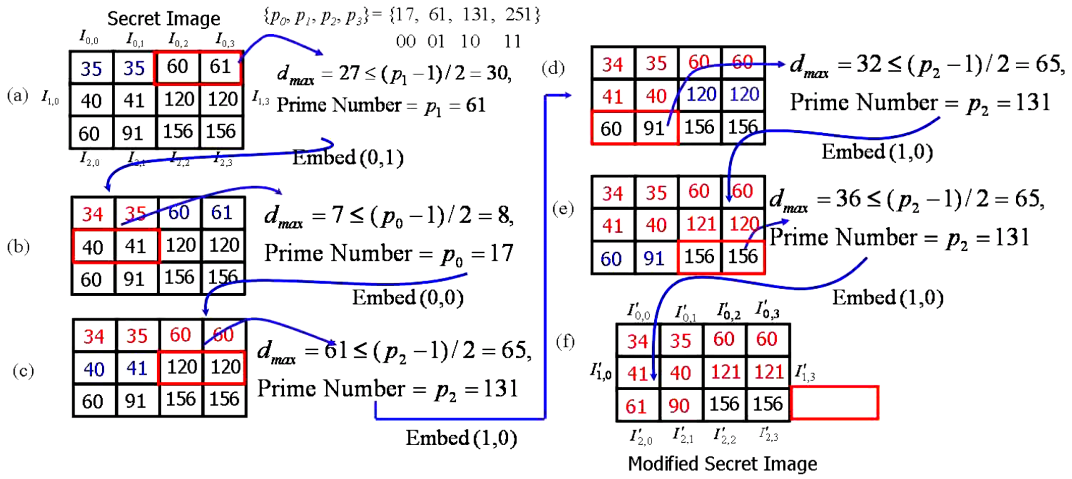


FIGURE 7. The way to produce the modified secret image in the proposed scheme

(a) Lena (b) F16 (c) Baboon

FIGURE 8. Three test images

In the recovering phase, two variables f'_0 and f'_1 can be extracted from the coecients of the reconstructed polynomial function. After that, two pixels $I'_{i,j}$ and $I'_{i,j+1}$ in the modified secret image can be recovered through f'_0 and f'_1 by using the formulas as follows:

$$\begin{cases} I'_{i,j} = (f'_0 - (p_j - 1)/2) + MED\left(I'_{i-1,j-1}, I'_{i-1,j}, I'_{i,j-1}\right) \\ I'_{i,j+1} = (f'_1 - (p_j - 1)/2) + MED\left(I'_{i-1,j}, I'_{i-1,j+1}, I'_{i,j}\right) \end{cases} \quad for \;\; j \in 0, 1, 2. \tag{15}$$

$$\begin{cases} I'_{i,j} = f'_0 \\ I'_{i,j+1} = f'_1 \end{cases} \quad for \;\; j \in 3. \tag{16}$$

4. **Experimental results.** This section introduces experimental results to demonstrate the effectiveness of the proposed scheme, in which a set of test gray-scale images are used. Fig. 8 shows the set of test images, including Fig. 8(a) Lena, Fig. 8(b) Jet, and Fig. 8(c) Baboon. Each image size is 512×512 pixels.

Moreover, the image quality is evaluated by using the peak signal-to-noise ratio ($PSNR$) to measure the distortion between the original image and recovered image. The $PSNR$ is evaluated as

$$PSNR = 10 \times \log_{10}\left(\frac{255^2}{MSE}\right) \tag{17}$$

The mean square error (MSE) is dened as follows:

$$MSE = \frac{1}{M \times N} \sum_{i}^{M} \sum_{j}^{N} \left(p_{i,j} - p'_{i,j}\right)^2, \tag{18}$$

where (M × N) is the size of an image; $p_{i,j}$ is the original pixel value and $p'_{i,j}$ is the recovered pixel value.

There are two different prime number sets used in our experiments. The prime number set I is {17, 61, 131, 251} which is also used in Yang *et al.*'s scheme. Fig. 9 and Fig. 10 summarize the experimental results using the proposed (2, 4)-threshold user-friendly image sharing with prime number set I. Fig. 9(a) display the recovered images using any two of four shadow images in Figs. 9(d)-(g). Fig. 9(b) displays the recovered images using any two of four shadow images in Figs. 9(h)-(k). Fig. 9(c) shows the recovered images when using any two of four shadow images in Figs. 9(l)-(o).

TABLE 1. The $PSNR_s$ values of the recovered images based on different user-friendly image-sharing schemes

	Thien et al. [7]	Yang et al. [10]	Proposed scheme
Lena	37.37	50.53	51.17
Jet	39.19	49.76	51.14
Baboon	34.75	49.17	51.15

Table 1 summarizes the image quality of the recovered images in a (2, 4)-threshold user-friendly image sharing manner compared with two previously published user-friendly image sharing schemes. The values in Table 1 are the $PSNR$ values of the recovered images. It reveals that the proposed scheme can reconstruct the secret image with high quality.

In Yang *et al.*'s scheme, the prime number used in the current block only depends on the pixel in the previous block. Thus, for a non-smooth block (i.e. edge block), this scheme has high probability of using Formula (4). For example, there exists a vertical edge in Fig. 10(a). According to Yang *et al.*'s scheme, the prime numbers indicated in Fig. 10(a) should be p3. This means Yang *et al.*'s scheme must use Formula (4) to produce f_0 and f_1. Note that using Formula (4) in Yang *et al.*'s scheme must also modify the second LSB (i.e. 7^{th} bit) of the secret pixel to embed the LSB of the pixel value difference as described in Subsection 2.1. This kind of modications will cause image degradation. On the other hand, the proposed scheme uses JPEG-LS median edge predictor to predict pixel value. Owing to the function of JPEG-LS median edge predictor, the proposed scheme rarely uses large prime number such as p_3 in Fig. 10(b). Moreover, the proposed scheme modies Formula (4) to Formula (14) such that the second LSB (i.e. 7^{th} bit) of the secret pixels needs not to be modied. Therefore, the proposed scheme can recover the secret image with high quality.

TABLE 2. Comparison of the proposed scheme with others in terms of the average $PSNR_s$ value of expanded shadow images

	Thien et al. [7]	Yang et al. [10]	Proposed scheme
Lena	24.80	23.32	27.69
Jet	25.65	23.14	26.91
Baboon	20.55	18.52	22.21

Table 2 lists the average $PSNR$ values between the expanded images and the original image by using the proposed scheme and other schemes. According to this table, the $PSNR$ values of shadow images of the proposed scheme are higher than those of the schemes in [7] and [10]. The reason is that applying JPEG-LS median edge predictor to predict pixel value will lead us to use small prime numbers to share secret pixels. As

described in Subsection 2.1, the shadow pixel $\hat{P}_i \ mod \ p_j = S(x_i)$ is obtained by locating its value as close to the average pixel value of the current block as possible, but the value also satisfies the constraint . Using small prime numbers make it possible to obtain shadow pixels with small pixel value difference between the shadow pixels and the average pixel value. Therefore, the shadow images produced by the proposed scheme have higher quality than those produced by [7] and [10].

5. **Conclusions.** This work presents a user-friendly system based on the use of JPEG-LS median edge predictor to determine the prime number for each block. According to the experimental results, the qualities of both the reconstructed secret image and shadow images in the proposed scheme are better than those in Yang *et al.*'s scheme. Additionally, the size of each shadow image is smaller than that of the original secret image. This feature benets the users when attempting to identify a large number of shadow images. More than well organized in terms of storage space, the small size of the shadow images makes it effcient for transmission.

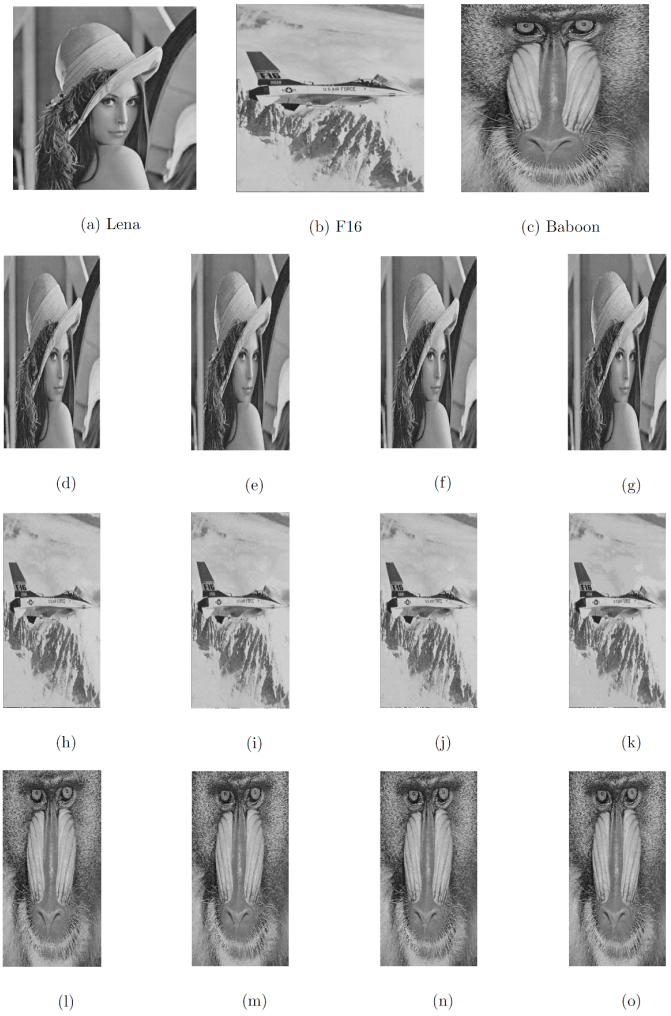
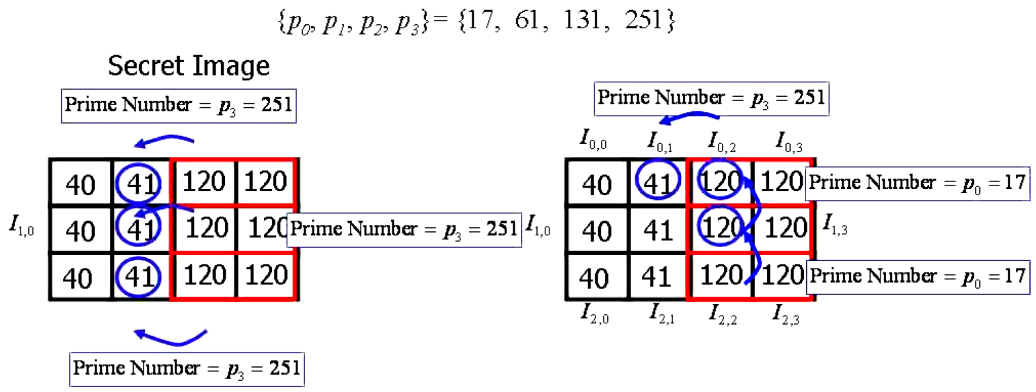


FIGURE 9. Results of the (2, 4)-threshold proposed scheme. (a) Reconstructed image Lena with PSNR 51.17dB, (b) Reconstructed image F16 with PSNR 51.14dB, (c) Reconstructed image Baboon with PSNR 51.15dB, (d)-(g) four shadow images for Lena image, (h)-(k) Four shadow images for Jet image, (l)-(0) Four shadow images for Baboon image.



(a) Yang *et al.*'s method (b) The proposed method

FIGURE 10. The prime numbers used in the edge block

REFERENCES

[1] G. R. Blakley, Safeguarding cryptographic keys, *Proc. National Computer Conference*, vol. 48, pp. 313-317, 1979.

[2] S. Bedi, E. A. Edirisinghe and G. Grecos, Improvement to the JPEG-LS prediction scheme, *Journal of Image and Vision Computing*, vol. 22, pp. 9-14, 2004.

[3] S. K. Chen and J. C. Lin, Fault-tolerant and progressive transmission of images, *Journal of Pattern Recognition*, vol. 28, pp.2466-2471, 2005.

[4] J. Jiang, B. Guo, and S. Y. Yang, Revisiting the JPEG-LS prediction scheme, *IEE Proceedings-Vision, Image and Signal Processing*, vol. 147, pp. 575-580, 2003.

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

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

[7] C. C. Thien and J. C. Lin, An image-sharing method with user-friendly shadow images, *IEEE Trans. Circuits and Systems for Video Technology,*, vol. 13, pp. 1161-1169, 2003.

[8] C. Y. Yang, W. C. Hu, and C. H. Lin , Reversible data hiding by coecient-bias algorithm, *Journal of Information Hiding and Multimedia Signal Processing*, vol. 1, no. 2, pp. 91-100, 2010.

[9] C. Y. Yang, C. H. Lin, and W. C. Hu, Reversible data hiding by adaptive IWT-coeRcient adjustment, *Journal of Information Hiding and Multimedia Signal Processing,*, vol. 2, no. 1, pp. 24-32, 2011.

[10] C. N. Yang, K. H. Yu, and R. Lukac, User-friendly image sharing using polynomials with different primes, *International Journal of Imaging Systems and Technology,*, vol. 17, pp. 40-47, 2007.