Journal of Network Intelligence
Taiwan Ubiquitous Information
 ISSN 2414-8105 (Online)
Volume **3**, Number **3**, August **2018**

Distortion Free Progressive BTC based Secret Image Sharing

Chin-Chen Chang

Department of Information Engineering and Computer Science,
Feng Chia University, Taichung 40724, Taiwan, R.O.C
Department of Computer Science and Information Engineering,
Asia University, Taichung 41354, Taiwan
alan3c@gmail.com

Yi-Ping Chiu

Department of Computer Science and Information Engineering,
National Chung Cheng University, Chia-Yi 62102, Taiwan, R.O.C
evapean@gmail.com

Chia-Chen Lin

Department of Computer Science and Information Management,
Providence University, Taichung 43301, Taiwan, R.O.C
Corresponding Author: mhlin3@pu.edu.tw

Yi-Hui Chen

Department of Applied Informatics and Multimedia,
Asia University, Taichung 41354, Taiwan, R.O.C
chenyh@asia.edu.tw

Received Jan 2018; revised April 2018

Abstract. *Following the development of a method by Blakley in 1979 to share secret data among participants, in 2008, Tso extended that method to transmit information while reducing the storage space and bandwidth by quantizing the secret image. Owing to its use of quantization to generate shares, the method of Tso fails to obtain a reconstructed image distortion free. In 2010, Ulutas et al. proposed a solution to obtain the reconstructed image entirely. However, their method incurs the share expansion problem. Therefore, this work presents a novel method that combines the methods of Tso and Ulutas et al. to transmit a secret image with small shares and obtains the reconstructed image distortion free. The proposed scheme also provides the progressive data transmission feature.*
Keywords: (K, n)-threshold secret sharing; Block truncation coding (BTC); Progressive data transmission

1. **Introduction.** Information security is of priority concern in transmitting data such as text, image audio or video, online due to the easy and public access nature of the Internet. Two conventional methods to protect a secret image during transmission over the Internet are cryptographic methods and steganography. Cryptographic methods transform a secret image into an unreadable format via an encryption algorithm (i.e., DES [17, 18, 21] or RSA [19, 20, 22]) pre-shared between the sender and the receiver. Encrypted images often draw the attention of malicious attackers during transmission owing to its unreadable format. Steganography conceals a secret image in a cover image. The cover image

can be any image retrieved online. Even a cover image hides a secret image and, then, a cover image is transformed into a stego-image. The stego-image is still nearly the same as the cover image for a human visual system; attackers are thus not easily suspicious of cover images even when they carry secret images. However, both conventional methods, cryptographic methods and steganography fail to guarantee a secret image that can be reconstructed once the encrypted format or stego-image has been modified or corrupted during data transmission [5], explaining the increasing popularity of the secret sharing method in the recent decade. Secret sharing divides a secret image into n sub-images called shares and, then, the reconstructed secret image can be obtained as long as at least $k(k \leq n)$ shares are collected. Restated, a secret image cannot be reconstructed with less than k shares. Due to such an attribute, the secret sharing method is also called (k, n)-threshold secret sharing.

Shamir [1] and Blakley [2] developed (k, n) threshold secret sharing schemes individually in 1979. The method of Shamir [1] shares secret data among n different participants using the polynomial approach [1]. Namely, the secret data is constructed degree polynomial with a constant. Other coefficients of the polynomial function are randomly selected in the range of $[0 \sim p)$, where p is a prime number. Shares are evaluated with respect to the values of the polynomial to be distributed to participants.

During the sharing phase, each participant obtains shares from the dealer. During the reconstruction phase, secret data is then reconstructed by any k or more shares from the participants. Secret data can not be reconstructed by any less than k shares. The method of Blakley shares secret data among n participants by using the geometric approach [2]. That method assumes that the secret data is a point in k-dimensional space. The point generates shares to be distributed by hyperplane equations. The coordinate of this point, which is used to represent the secret, can be revealed by intersecting the hyperplanes. However, size of the share images is normally larger than or equal to the size of the secret image when the above two methods are applied to image applications. In 2002, Thien and Lin used the method of Shamir to protect secret images [3]. The method of Thien and Lin first selects a prime number p as 251, because 251 is the largest prime number and less than for 8-bit depth gray-level images. If a pixel value in a secret image is greater than 250, this value is truncated to 250 before the sharing phase. Following truncation, a secret image is divided into some blocks with k pixels. A polynomial with $k-1$ degree is constructed using each block. The polynomial values are evaluated to generate the corresponding values of a share. An image with a size of $N \times M$ is partitioned into share images with a size of $(N \times M)k$.

In 2008, based on the secret sharing concept of Blakley, Tso attempted to reduce the storage space and bandwidth by quantizing a secret image [4]. However, the quantization process causes distortion during the reconstruction phase. To avoid such distortion, Ulutas et al. proposed a solution for the method of Tso in 2011 [5]. The method of Ulutas et al. evaluates the distortion between the secret image and quantized image first and, then, records the above distortion by using a difference image. Finally, Ulutas et al. generated share images by using quantized image and difference image. Comparing the methods of Tso and Ulutas et al. reveals that the quantized image is partitioned into k pixel groups in the method of Tso, explaining why the size expansion ratio is $1/k$. In contrast, the size expansion ratio of the method of Ulutas et al. is up to $2 \times (1/k)$ owing to the difference image.

In addition to share expansion, progressive transmission has also received considerable attention recently [6, 14, 15, 16]. When an image is transmitted to a receiver over the Internet via conventional secret sharing methods, receivers must wait until end of the transmission before they can successfully view the entire image. With a lower bandwidth

or a larger image, receivers must spend additional waiting time in viewing the received image, implying that decision making is delayed until all transmission data have been received. In 2001, Chen and Lin designed a progressive image transmission method that applies the concept of image sharing with fault-tolerant capability [11]. Via conventional progressive image sharing, the information of an image is partitioned into several parts, and the partial data are transmitted in a progressive manner. A receiver can obtained the reconstructed image more clearly by gathering more partial data during the progressive image sharing phase.

Consider a situation in which both distortion free and transmission speed are important in certain applications e.g., military and medical. By using the progressive method to transmit a secret image, only a portion of data of the original information is transmitted and, then, more data is gathered at the receiver side by more rounds. Finally, a reconstructed image can be obtained by collecting data transmitted at each round. In this case, if requiring data urgently, a receiver uses partial received data to reconstruct the information with less image quality efficiently. Once all transmission rounds are completed, the receiver still derives a complete secret image with high image quality to assist further decisions. Since the progressive transmission works with the secret image, the security of a transmitted secret image can be protected from malicious attacks as the conventional secret image methods do.

The proposed scheme first divides a secret image into non-overlapping blocks by using the block truncation coding (BTC) method [8, 9, 10]. Once a difference image is obtained, shares are generated using the methods of Tso and Ulutas et al. Since the proposed scheme is characterized by progressive secret sharing without distortion, four additional values are transmitted along with the difference image in each transmission round. Only during the last round, five additional values are transmitted along with difference image. In the proposed scheme, the reconstructed image becomes clearer when the transmission round is increased. During the final round, the reconstructed image is distortion free.

The rest of this paper is organized as follows. Section 2 introduces BTC method and the schemes of Tso and Ulutas et al. Section 3 then describes the details of the proposed scheme. Next, Section 4 summarizes the experimental results. Conclusions are finally drawn in Section 5, along with directions for future research.

2. **Related works.**

The proposed scheme uses the BTC method [10] first to apply the methods of Tso and Ulutas et al. to progressive secret sharing. Based on quantization, the method of Tso decreases bandwidth and storage capacity of the share images during transmission [4]. However, the reconstructed image is distorted by the method of Tso. To generate a reconstructed image distortion free, Ulutas et al. developed a method to improve Tsos method [5]. This section describes these methods: BTC, the method of Tso and the method of Ulutas et al. The following subsections illustrate these methods in further detail.

2.1. **Block Truncation Coding.**

Block truncation coding (BTC), an image compression method, compresses a block with a size of $m \times m$, normally a block with 4×4 pixels. For instance, for an image with a size of 512×512, the image is divided into 4×4 non-overlapping block. Therefore, the image is divided into $(512 \times 512)/(4 \times 4) = 16384$ image blocks. The BTC compression is introduced as follows.

An image is divided into some blocks with a size of $m \times m$ and, then, the threshold for each block is computed, in which threshold is usually the average of all pixels in a block.

Next, all pixels are compared with the threshold to construct a bit map for each block. For a pixel value higher than the threshold for each block, the corresponding address of the bit map is 1; otherwise, the pixel of a bit map is 0. Additionally, two reconstruct values (i.e., avg_0 and avg_1) are generated to represent the block. The mean value of pixel values, which are higher than the threshold for each block is ; otherwise, the mean value of the pixel values is avg_1. Finally, the reconstructed values (avg_0 and avg_1) and the bitmap are transmitted to the receiver for each block.

2.2. **Method of Tso [4].** The method of Tso is based on Blakley's secret sharing. That method uses quantization method to decrease the bit depth of the image. Therefore, during the transmission, the required bandwidth of the share images and storage capacity are reduced. Let secret image denote $S = \{s_{i,j} | s_{i,j} \in [0 - 255], 1 \leq i \leq N, 1 \leq j \leq M\}$. Also, the quantized secret image is $S' = \left\{s'_{i,j} | s'_{i,j} = [s_{i,j}/b], s_{i,j} \in S, 1 \leq i \leq N, 1 \leq j \leq M\right\}$, while b represents a random integer.
Quantized secret image is divided into non-overlapping blocks with k pixels. A polynomial is constructed using each block, as shown in (1).

$$(t_n + a_1 x_1 + a_2 x_2 + \cdots + a_{k-1} x_{k-1}) mod\ p \equiv a_k, \tag{1}$$

where denote the pixel values of each block, and $p = [255/b]$. $x_1, x_2, , \ldots, x_k$ are randomly selected to obtain the shared values $t_1, t_2, \ldots, t_n$. Therefore, this method generates share images after the sharing stage with $(N \times M)/k$ pixels.

2.3. **Method of Ulutas et al. [5].** To generate a reconstructed image distortion free, Ulutas et al. developed a method to improve Tso's method. By using a difference image, their method records the difference between the original secret image and quantized image. The difference image of the same size as the secret image since each secret pixel is used to calculate a truncation error or difference during processing. Finally, share images are generated using the quantized secret image and difference image. A polynomial is also constructed using each block, as shown in (2).

$$\begin{gathered}(s'_{ij} x_1^z + s'_{ij+1} x_k^z + \cdots + s'_{ij+k-1} x_k^z) mod\ (\lfloor 255/b \rfloor) \equiv sh_{mn}^z, z \in \{1, \ldots, n\} \\ (d'_{ij} x_1^z + d'_{ij+1} x_k^z + \cdots + s'_{ij+k-1} x_k^z) mod\ (\lfloor 255/b \rfloor) \equiv sh_{mn}^z + 1, z \in \{1, \ldots, n\} \\ m = i, n = \lfloor j/k \rfloor \times 2 + 1\end{gathered} \tag{2}$$

where S denotes a secret image by $S = \{s_{ij} | s_{ij} \in [0 - 255], 1 \leq N, 1 \leq j \leq M\}$ and $S' = \left\{s'_{ij} | s'_{ij} \in [0 - (255/b)], 1 \leq N, 1 \leq j \leq M\right\}$, where b represents a random integer. Difference image is denoted by $D = \{d_{ij} | d_{ij} = s_{ij} - b \times \lfloor s_{ij}/b \rfloor, d_{ij} \in [0 - (b-1)], 1 \leq i \leq N, 1 \leq j \leq M\}$. $x_1^z, x_2^z, \ldots, x_k^z$ are randomly selected for each participant to generate the shared pixel values $sh_{mn}^z, z \in \{1, \ldots, n\}$.

Each block with k pixels from the difference and quantized images corresponds to a pair of shared values in each share image. Therefore, this method generates share images after the sharing stage with $2 \times (N \times M)/k$ pixels.

3. **Proposed Scheme.** The proposed scheme consists of two phases: share construction and revealing. During the share construction phase, an original secret image is divided into non-overlapping blocks sized 4×4 pixels by using the BTC method and, then, a reconstructed image is formed by using the decoding procedure of BTC. Comparing the difference between original image and reconstructed image leads to a difference image. Notably, the original image and difference image have the same size. Finally, the proposed scheme transmits the share images of difference image and also transmits some values to

the receiver for each round. The receiver can reconstruct the image with the share images of difference image and the values. At the end of transmission round, the receiver can obtain the reconstructed image distortion free.

3.1. **Method of Ulutas et al. [5].** The proposed scheme uses the BTC method to divide a secret image into several non-overlapping blocks with size of 4×4. The BTC method has been described earlier in subsection 2.1. This subsection describes the generated shares of a given secret image block. This part consists of three steps, as shown in Figure 1. The following phases describe the details of these phases.

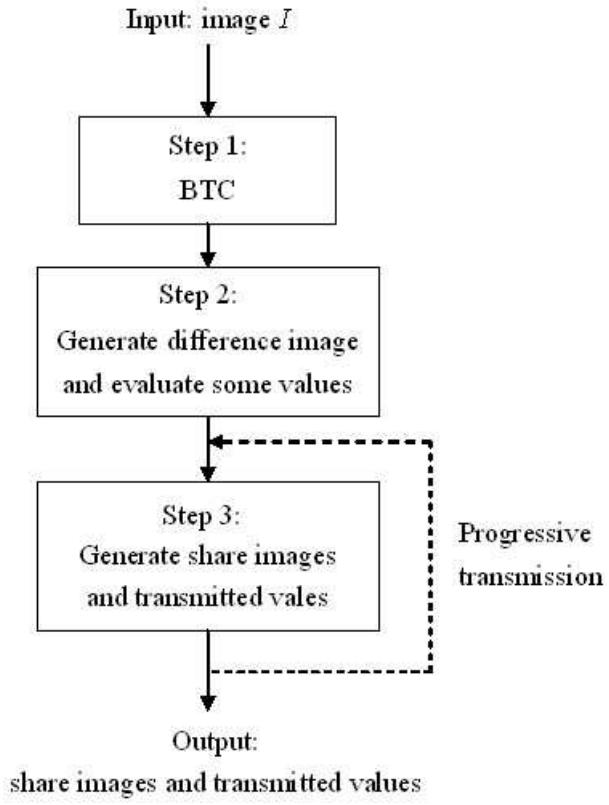


FIGURE 1. Diagram of the proposed three-stepped sharing procedure

Step 1: Perform BTC encoding.
As mentioned in subsection 2.1, the reconstructed values (avg_0 and avg_1) and a bitmap for each block of a secret image are generated by using the encoding procedure of the BTC method. Assume there is a block of the secret image, as shown in Figure 2. The reconstructed values (avg_0 and avg_1) and its bitmap are generated after BTC encoding.

Step 2: Generate a difference image and evaluate some values.
BTC encoding divides a secret image into $(N \times M)/(4 \times 4)$ non-overlapping blocks. Therefore, after BTC encoding, the reconstructed values(avg_0 and avg_1) and its bitmap are generated for each block. A decompression-map is then constructed using the reconstructed values (avg_0 and avg_1) and corresponding bitmap, as shown in Figure 3, Namely, a pixel value is set as avg_0 in decompression-map when its corresponding value is "0" in the bitmap. Otherwise, the pixel value is set as avg_1 in the decompression-map when the corresponding value is "1" in the bitmap. Next, two values, i.e. $bmap_0$ and $bmap_1$, are derived to represent the bitmap according to the following rules: the first 8 bits of a bitmap are collected to generate an integer value as $bmap_0$ and the following 8 bits of

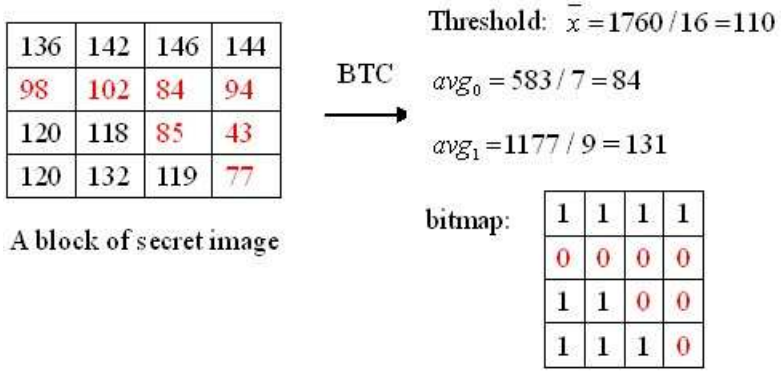


FIGURE 2. Example of BTC encoding for a block of a secret image

bitmap to generate an integer value as $bmap_1$,as shown in Figure 4. Figures 3 and 4 show an example of a block with 4×4, which follows the example shown in Figure 2.

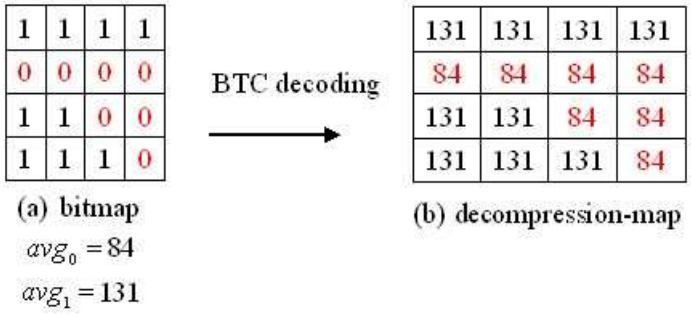


FIGURE 3. Example of BTC decoding results

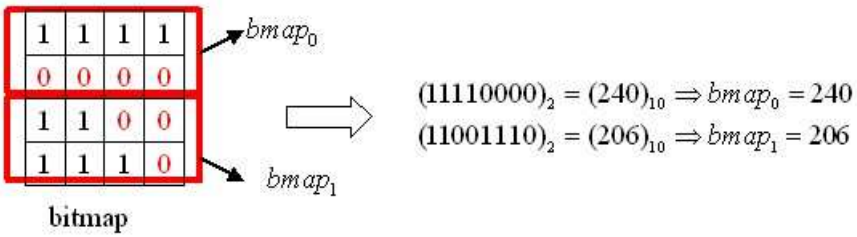


FIGURE 4. Example of generating two values to represent a given bitmap

The difference with the corresponding original block is then estimated using the decompression-map. The difference image is then derived by evaluating the difference between the original block and the decompression-map, i.e., the distortion is recorded in the difference image. Notably, the difference image is of the same size as the secret image. Figure 5 shows the example of difference image by differentiating between the original block and the decompression-map. Follow Figures 3 and 4, Figure 5 displays the example in which the secret image has a size of 4×4. To apply the progressive transmission, the partial values of avg_0 and avg_1 must be generated in advance to ensure their transmission in different rounds by using Equation (3). Additionally, avg_0 and avg_1 are divided S_{total} into partial-values t_0 and t_1. Moreover, r_0 denotes the remainder between avg_0 and S_{total}, r_1

and represents the remainder between avg_1 and S_{total}. Restated, a secret image is partitioned into non-overlapping blocks with a size of 4×4. For each block, the corresponding partial-values t_0 and t_1 are remainders, and r_0 and r_1 are obtained by Equation (3).

$$\begin{gathered} t_0 = avg_0 / S_{total}, \\ r_0 = avg_0 \% S_{total}, \\ t_1 = avg_1 / S_{total}, \\ r_1 = avg_1 \% S_{total}, \\ 2^{u-1} < avg_0 < 2^u, 2^{v-1} < avg_1 < 2^v, \\ u' = u + S_{total}, \\ v' = v + S_{total}, \end{gathered} \tag{3}$$

where S_{total} is the total rounds of transmission, and u, v, u', v' are integer numbers. (u, v, u', $v' > 0$). For the same example described above, give $S_{total} = 5$. Figure 6 illustrates how to generate the partial data of avg_0 and avg_1 each round.

5	11	15	13
14	18	0	10
-11	-13	1	-41
-11	1	-12	-7

Difference image

FIGURE 5. Example of the difference image

$t_0 = avg_0 / S_{total} = 84 / 5 = 16$
$t_1 = avg_1 / S_{total} = 131 / 5 = 26$
$r_0 = avg_0 \% S_{total} = 4$
$r_1 = avg_1 \% S_{total} = 1$
$2^{u-1} < avg_0 < 2^u, 2^{v-1} < avg_1 < 2^v$
$\Rightarrow 2^6 < 84 < 2^7, 2^7 < 196 < 2^8$
$\Rightarrow u' = 7 + 5 = 12, v' = 8 + 5 = 13$

FIGURE 6. Example of progressive values with $S_{total} = 5$

Step 3: Generate shares and transmitted values.
Shares and some transmitted values are generated for each block. After Step 2 in the construction phase, a difference image is generated by recording the distortion from the secret image. The size of the difference image is the same as that of the secret image. By using the methods of Tso and Ulutas et al., the proposed scheme generates shares for the difference image. Before shares of difference image are generated, all values in the difference image must be justified to be positive ones. Also, the proposed progressive secret sharing is implemented by calculating some values for each round. The details are described in the following steps.
Let D and S_{cur} denote the difference image and the current round, respectively, where

$S_{cur} = 1, 2, \ldots, S_{total}$. Assume that a secret image is $M \times N$. The following examples are simplified by setting the size of a secret image as 4×4 in the following steps.

Step 3.1: If $S_{cur} < S_{total}$, compute the two middle-difference images D" and D'. The middle-difference images D" and D' are computed using Equations (4) and (5), respectively.

$$D'' = \left\{ d''_{ij} | d''_{ij} = \lfloor d_{ij} / \left(S_{total} \times (S_{total} - S_{cur} + 1) \right) \rfloor, d_{ij} \in D, 1 \le i \le M, 1 \le j \le N \right\}. \tag{4}$$

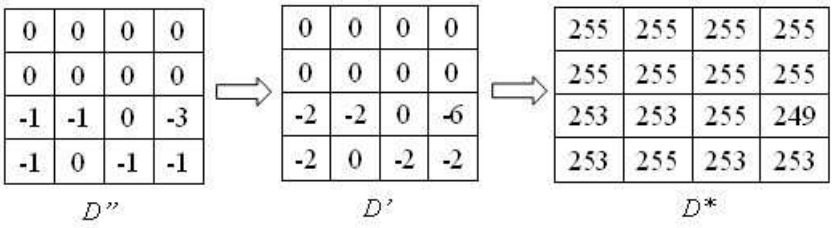
$$D' = \left\{ d'_{ij} | d'_{ij} = S_{cur} \times d''_{ij}, d''_{ij} \in D'', S_{cur} < S_{total}, 1 \le i \le M, 1 \le j \le N \right\}. \tag{5}$$

Middle -difference image D' is generated by Equations (4) and (5). To avoid negative numbers in D', the 255 are added for all pixel values in D' as shown in Equation (6). Later, the pixel values must be verified as to whether they are greater than 255. This is owing to that the range of pixel values is 0 to 255 for 8-bit depth gray level images. Let D^* denote the transmitted-difference image of adjusting D'. Equation (7) shows the check rule.

$$D^* = \left\{ d^*_{ij} | d^*_{ij} = d_{ij} + 255, d'_{ij} \in D', 1 \le i \le M, 1 \le j \le N \right\}. \tag{6}$$

$$All pixel values in D^* must be verified : if d^*_{ij} > 255 then d^*_{ij} = 255. \tag{7}$$

The transmitted-difference image D^* is then generated by Equations (6) and (7). The shares are then constructed using D^*, with the details described in Step3. Figure 7 illustrates the example with $S_{cur} = 2$ for Step 3.1.



where $D'' = \{ d''_{ij} \mid d''_{ij} = \lfloor d_{ij} / 20 \rfloor, d_{ij} \in D, 1 \le i \le 4, 1 \le j \le 4 \}$,

$D' = \{ d'_{ij} \mid d'_{ij} = 2 \times d''_{ij}, d''_{ij} \in D'', 1 \le i \le 4, 1 \le j \le 4 \}$,

$D^* = \{ d^*_{ij} \mid d^*_{ij} = d'_{ij} + 255, d'_{ij} \in D', 1 \le i \le 4, 1 \le j \le 4 \}$

FIGURE 7. Example of Step3.1 with $S_{total} = 5$ and $S_{cur} = 2$

Step 3.2: If $S_{cur} = S_{total}$, compute the middle-difference images D', and find the largest negative value as in order to adjust pixel values in difference image D' to be positive values. The middle-difference image D' and the transmitted-difference image $D^a st$ are computed using Equations (8) and (9), respectively.

$$D' = \left\{ d'_{ij} | d'_{ij} = d_{ij}, d_{ij} \in D, 1 \le i \le M, 1 \le j \le N \right\}. \tag{8}$$

$$D^* = \left\{ d^*_{ij} | d^*_{ij} = d'_{ij}, d'_{ij} \in D', 1 \le i \le M, 1 \le j \le N \right\}. \tag{9}$$

The transmitted-difference image D^* is generated by Equations (8) and (9), and, then, the shares are constructed using D^*. Figure 8 displays the example with $S_{cur} = S_{total} = 5$ for Step 3.2.

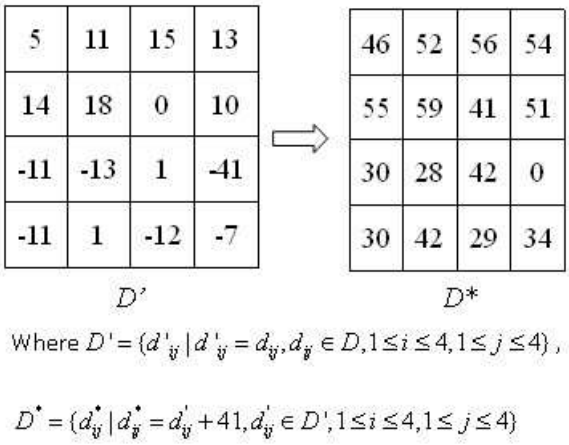


FIGURE 8. Example of Step 3.2 with $S_{cur} = S_{total} = 5$ and $|m_d| = 41$

Step 3.3: Generate shares of the transmitted-difference image D^* by using the method of Tso.
The proposed scheme does not use the quantization method to generate shares. Therefore, the transmitted-difference image D^* is constructed using the polynomial function, as shown in Equation (10).

$$(t_n + a_1 x_1 + a_2 x_2 + \cdots + a_{k-1} x_{k-1}) \, mod \, p \equiv a_k, \tag{10}$$

where p represents a prime number. Notably, $p = 251$ is selected in this scheme since 251 is the largest prime number and less than 2^8 for 8 bit-depth gray level images. Additionally, $a_1, a_2, \ldots, a_k$ represent the pixel values of D^*, and $x_1, x_2, \ldots, x_k$ are randomly selected to obtain the shared values $t_1, t_2, \ldots, t_n$. Figure 9 shows an example of generating shares by (2, 2) threshold with $x_1 = 1$ and $x_2 = 0$ for Step 3.3.

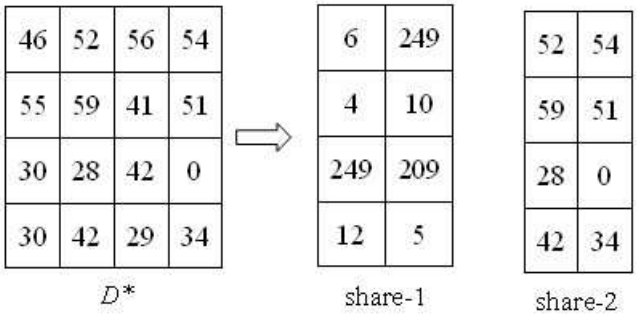


FIGURE 9. Example of generating shares using (2, 2) threshold for Step 3.3 with $x_1 = 1$ and $x_2 = 0$

Step 3.4: Evaluate two transmitted-values for progressive secret sharing.
Use six values generated during Step 2: t_0, t_1, r_0, r_1, u' and v' to compute two values q_0 and q_1 by Equations (11) and (12), respectively.

$$q_0 = t_0 - u' \times (S_{total} - S_{cur}), S_{cur} \le S_{total}, \tag{11}$$

$$q_1 = t_1 - v' \times (S_{total} - S_{cur}), S_{cur} \leq S_{total}, \tag{12}$$

where q_0 and q_1 are estimated by Equations (11) and (12), respectively. Next, check q_0 and q_1 according to the following conditions:

If $q_0 < 0$ then $q_0 = 1$.
If $q_1 < 0$ then $q_1 = 1$.

Finally, transmitted-values v_0 and v_1 are calculated by $v_0 = S_{cur} \times q_0 + r_0$ and $v_1 = S_{cur} \times q_1 + r_1$, respectively. According to the same example shown in Step 2, we know that $t_0 = 16, t_1 = 26, r_0 = 4, r_1 = 1$ and $u' = 12, v' = 13$ with $S_{cur} = 2$. Therefore, $q_0 = 1612 \times (52) < 0$ is used to get the result of $q_0 = 1$, and $q_1 = 2613 \times (52) < 0$ is to get the result of $q_1 = 1$. Finally, two transmitted-values v_0 and v_1 can be computed by q_0 and q_1 as $v_0 = 2 \times 1 + 4 = 6$ and $v_1 = 2 \times 1 + 1 = 3$.

Step 3.5: Transmit the four values.
Four values $bmap_0, bmap_2, v_0, v_1$ and shares of D^* are transmitted in each round. Notably, when $S_{cur} = S_{total}$, the adjusted value $|m_d|$ must be transmitted as well to obtain the reconstructed image with distortion.
Step 3.3 reveals that each k pixel of D^* generates a pixel of a share. The proposed scheme also transmits the transmission values $bmap_0, bmap_2, v_0, v_1$in each round except for the last round, and $(M \times N)/(4 \times 4)$ blocks are generated for a secret image sized $(M \times N)$. Therefore, the proposed scheme generates a share image with the size of $(M \times N)/k + T \times (M \times N)/(4 \times 4)$, where $T = 4$ if $S_{cur} < S_{total}, T = 5$ if $S_{cur} = S_{total}$. Finally, the size expansion ratio is $\frac{\frac{M \times N}{k} + T \times \frac{M \times N}{4 \times 4}}{M \times N} = \frac{1}{k} + \frac{T}{4 \times 4}$.

3.2. **Revealing Phase.** The revealing phase comprises five steps to reconstruct the secret image of each block for each round. Notably, $T \times \frac{M \times N}{4 \times 4}$ transmitted values are transmitted along with shares in each round. The transmitted values contain three data types: the values of $bmap_0, bmap_1$ are used to reconstruct bit map. The values of v_0 and v_1 are used to reconstruct decompression-map. The shares are used to reconstruct the transmitted-difference image D^*. Figure 10 shows five corresponding steps, as described in following paragraphs.

Step 1: Construct the bitmap for each block by using two received values $bmap_0, bmap_1$.
.

By using $bmap_0 = 240, bmap_1 = 206$, two bit strings that generate a bitmap: $(240)_{10} = (11110000)_2, (206)_{10} = (11001110)_2$ can be derived as shown in Figure 11.
Step 2: Reconstruct the decompression-map by using the bitmap obtained from Step 1 and the two received transmitted-values v_0 and v_1. Pixel value in the decompression-map is represented as if its corresponding bit in the bitmap is "0"; otherwise, the pixel value in decompression-map is represented as v_1. Figure 12 illustrates an example of reconstructing the decompression-map.

Step 3: Use received shares to reconstruct transmitted-difference image D^* by using the polynomial shown in Equation (10). Step 4: Compute middle-difference images D' by using D^* and $|m_d|$ according to the following two cases:

Case1: $S_{cur} < S_{total}, D' = \left\{ d'_{ij} | d'_{ij} = d^*_{ij} - 255, d^*_{ij} \in D^*, 1 \leq i \leq M, 1 \leq j \leq N \right\}$.
Case2: $S_{cur} = S_{total}, D' = \left\{ d'_{ij} | d'_{ij} = d^*_{ij} - |m_d|, d^*_{ij} \in D^*, 1 \leq i \leq M, 1 \leq j \leq N \right\}$.

Figures 13 and 14 illustrate an example of computing the middle-difference images D' with $S_{cur} = 2$ and $S_{cur} = 5$, respectively.

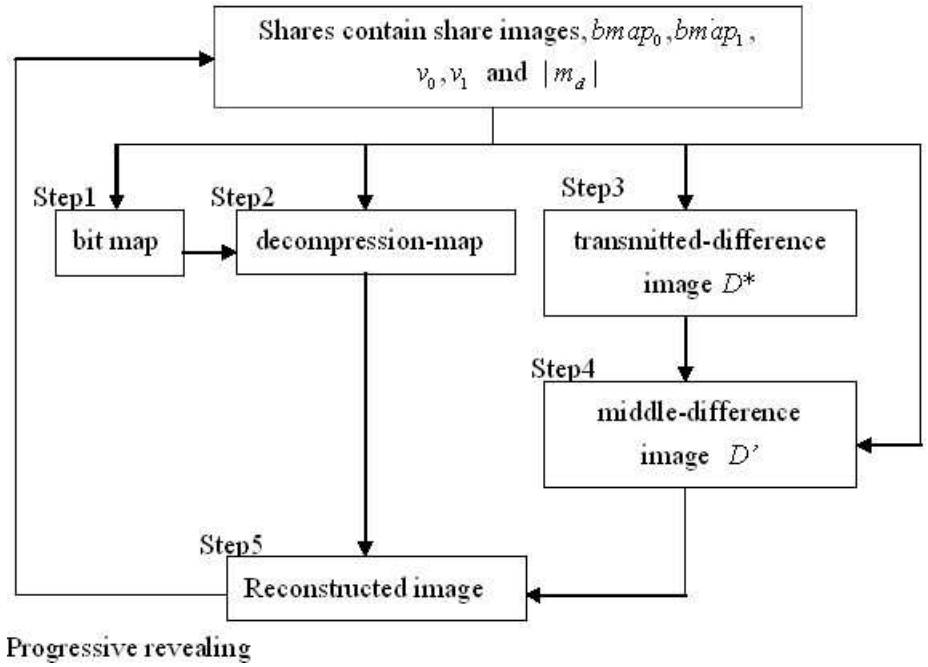


FIGURE 10. Diagram of the revealing phase with five steps

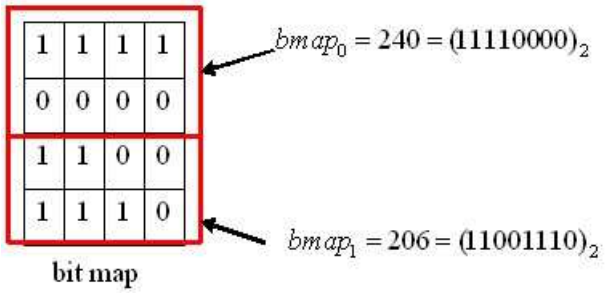


FIGURE 11. Example of reconstructing the bitmap

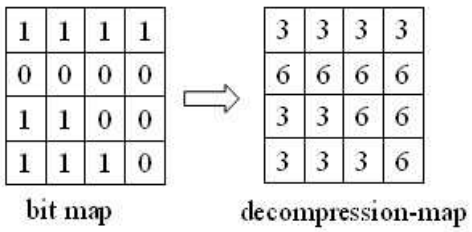


FIGURE 12. Example of reconstructing the decompression-map with $v_0 = 6$ and $v_1 = 3$

Step 5 : Sum up the decompression-map and D' to generate the reconstructed image. Figure 15 illustrates an example of reconstructing the image with $S_{cur} = 2$. During the final round, the transmitted-values v_0 and v_1 are equivalent to avg_0 and avg_1, respectively; in addition, the receiver obtains the $|m_d|$ value as well. Therefore, the receiver can reconstruct the distortion free image. Figure 16 shows an example.

Notably, the reconstructed image shown in Figure 16 is the same as original secret image shown in Figure 2.

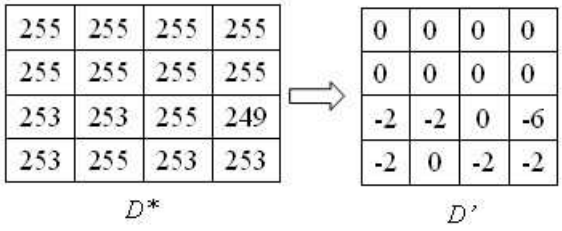


FIGURE 13. Example of reconstructing middle-difference images D' with $S_{cur} = 2$, where $D' = \left\{d'_{ij} | d'_{ij} = d^*_{ij} - 255, d^*_{ij} \in D^*, 1 \le i \le 4, 1 \le j \le 4\right\}$

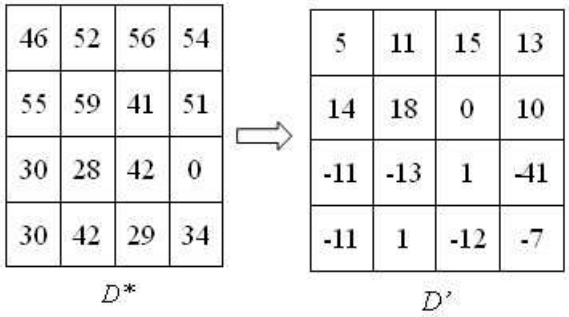


FIGURE 14. Example of reconstructing middle-difference images D' with $S_{cur} = 5$ and $|m_d| = 41$, where $D' = \left\{d'_{ij} | d'_{ij} = d^*_{ij} - |m_d|, d^*_{ij} \in D^*, 1 \le i \le 4, 1 \le j \le 4\right\}$

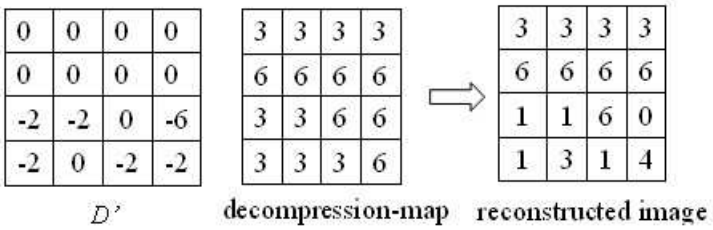


FIGURE 15. Example of reconstructing an image with $S_{cur} = 2$

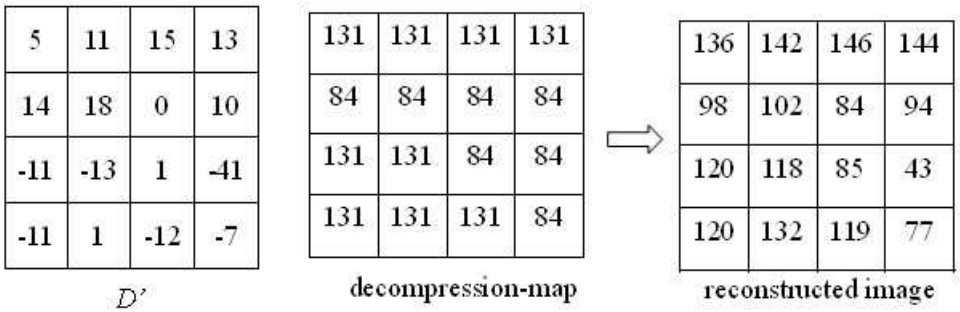


FIGURE 16. Example of reconstructing an image with $S_{cur} = 5$

4. **Experimental Results.** Our results demonstrate that the proposed scheme provides progressive transmission and distortion free capability for the secret sharing scheme. This section summarizes the experimental results of the proposed scheme for (2, 2) case with 5, 10 and 15 rounds, respectively. Experiments are performed by coding the algorithm

in C++. The test images are three gray level images "Lena," "Baboon," and "Peppers" with a size of 512×512 as shown in Figure 17.

FIGURE 17. (a)-(c) are the secret images of "Lena" , "Baboon", and "Peppers", respectively

Image quality of the reconstructed image is evaluated using the peak signal-to-noise ratio (PSNR). A PSNR with an infinity value of reconstructed image implies that the difference between secret image and reconstructed image is zero. Restated, no distortion occurs for a reconstructed image with infinity value of PSNR. PSNR is defined by Equation (13), where p_{ij} and q_{ij} denote the pixel values of a reconstructed image and secret image, respectively. Also, MSE q_{ij} denotes the mean square error between the reconstructed image and secret image.

$$\begin{aligned} PSNR &= 10 \times log_{10}\left(\frac{255^2}{MSE}\right) dB, \\ MSE &= \left(\frac{1}{M \times N}\right) \sum_{i=1}^{N} \sum_{j=1}^{M} (p_{ij} - q_{ij})^2 \end{aligned} \tag{13}$$

Figure 17 shows the tested images. Figures 18-20, Figures 21-23, and Figures 24-26 show the reconstructed images in each round with $S_{total} = 5$, $S_{total} = 10$, and $S_{total} = 15$ for the three tested images, respectively. Tables 1, 2 and 3 list the PSNRs of the reconstructed images with $S_{total} = 5, S_{total} = 10$, and $S_{total} = 15$, respectively.

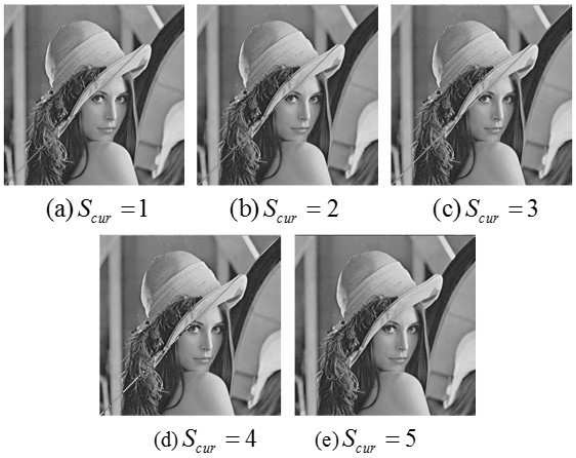


FIGURE 18. (a)-(e) present the reconstructed images of Lena for each round with $S_{total} = 5$ by the proposed scheme

Figures 17-20 reveal that although the reconstructed images in the first four rounds are damaged, the reconstructed images in the final round are the same as the original images. Table 1 reveals that the corresponding PSNRs of reconstructed images in the first four rounds are less than 21 dB. In contrast, PSNRs of the reconstructed images listed in Table 1 indicate that three reconstructed images are the same as the original images.

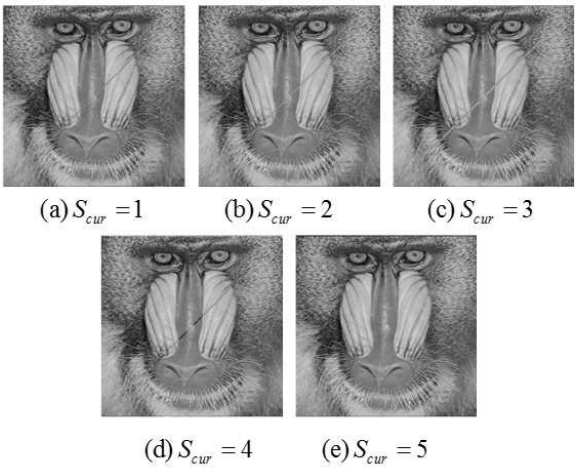


FIGURE 19. (a)-(e) describe the reconstructed images of Baboon for each round with $S_{total} = 5$ by the proposed scheme

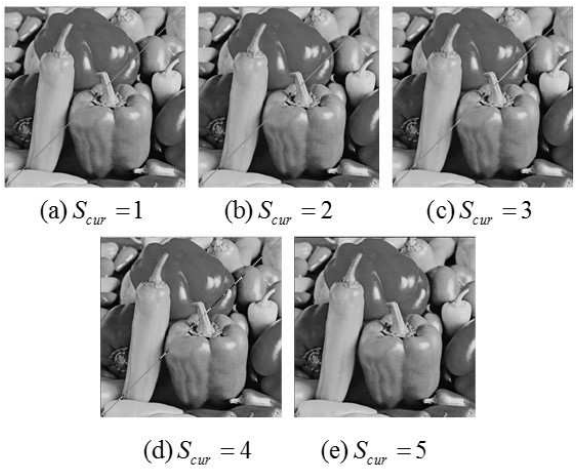


FIGURE 20. (a)-(e) describe the reconstructed images of Peppers for each round with $S_{total} = 5$ by the proposed scheme

TABLE 1. PSNRs of reconstructed images with $S_{total} = 5$ for the three tested images in each round

S_{total}	1	2	3	4	5
Lena	17.7790	17.8774	18.4711	19.5061	∞
Baboon	18.0592	18.2211	19.0491	20.0728	∞
Peppers	18.1037	18.2109	18.7064	19.6137	∞

Since the method of Ulutas et al. generates shares by using the difference image and quantized secret image, their size expansion ratio is $2/k$. The size expansion ratios of the method of Tso and the proposed scheme are $1/k$ and $\frac{1}{k} + \frac{T}{4\times4}$, respectively. Therefore, shares generated by the proposed scheme and that of Tso are smaller than that of Ulutas et al. As for distortion free capability, both the method of Ulutas et al. and the proposed method have distortion free reconstructed images.

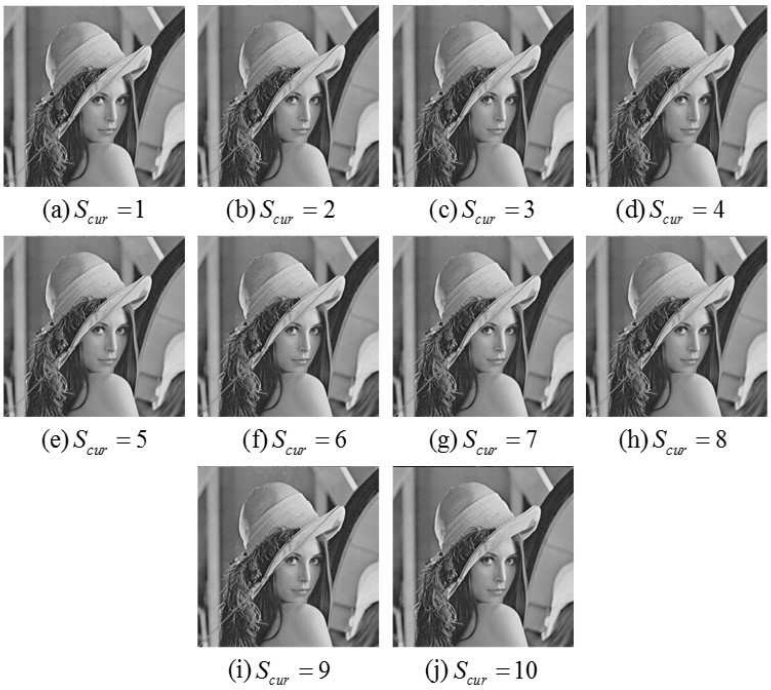


FIGURE 21. (a)-(j) represent the reconstructed images of "Lena" for each round with $S_{total} = 10$ by the proposed scheme

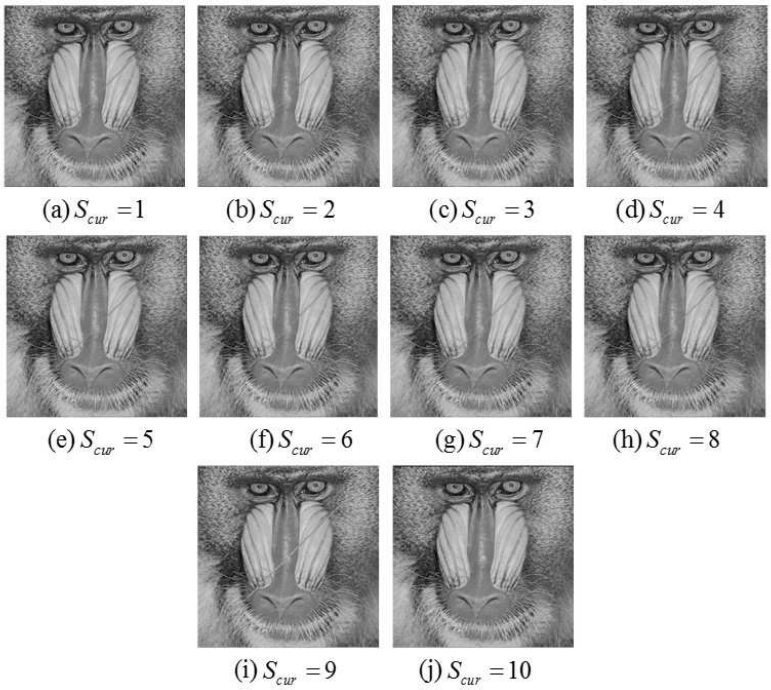


FIGURE 22. (a)-(j) represent the reconstructed images of "Baboon" for each round with $S_{total} = 10$ by the proposed scheme

5. **Conclusions.** The proposed scheme combines BTC method and the methods of Tso and Ulutas et al. to achieve distortion free capability and progressive secret sharing with small shares simultaneously. Based on the comparisons presented in Section 4, the proposed scheme also supports (k,n)-threshold secret sharing. Moreover, the proposed

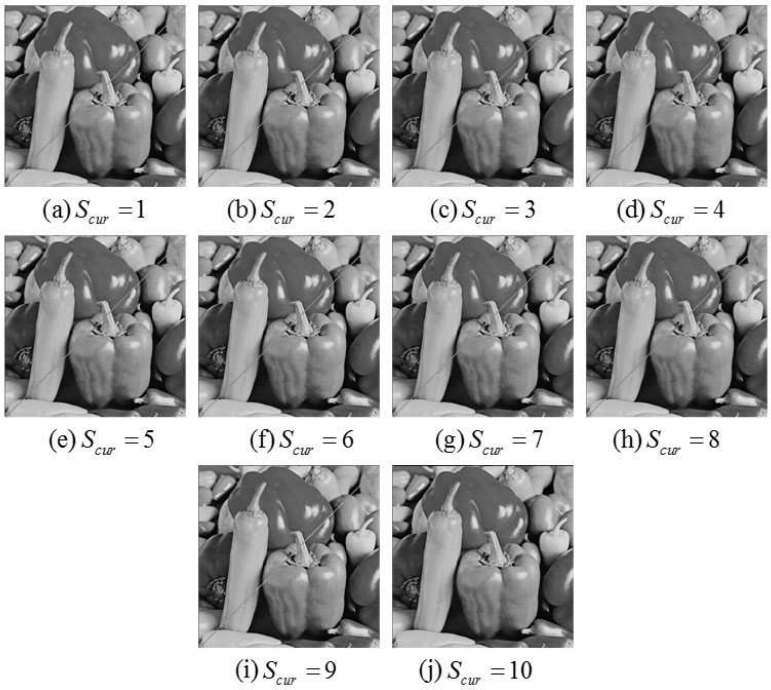


FIGURE 23. (a)-(j) represent the reconstructed images of "Peppers" for each round with $S_{total} = 10$ by the proposed scheme

TABLE 2. PSNRs of reconstructed images with $S_{total} = 10$ for the three tested images in each round

S_{cur}	1	2	3	4	5
Lena	17.8357	17.8498	17.8652	17.8805	19.6687
Baboon	18.1103	18.1241	18.1508	18.1713	20.0269
Peppers	18.1666	18.1818	18.2008	18.2162	19.8559
S_{cur}	6	7	8	9	10
Lena	19.6900	19.7127	19.7336	19.9659	∞
Baboon	20.0481	20.0706	20.1187	20.5772	∞
Peppers	19.8781	19.9037	19.9288	20.2445	∞

TABLE 3. PSNRs of the reconstructed images with $S_{total} = 15$ for the three tested images in each round

S_{cur}	1	2	3	4	5
Lena	17.8932	17.9072	17.9213	17.9353	19.7815
Baboon	18.1758	18.1895	18.2032	18.2170	20.1522
Peppers	18.2210	18.2362	18.2515	18.2667	19.9411
S_{cur}	6	7	8	9	10
Lena	19.8029	19.8244	19.8459	19.8675	19.8891
Baboon	20.1752	20.1967	20.2181	20.2396	20.2612
Peppers	19.9635	19.9876	20.0101	20.0326	20.0551
S_{cur}	11	12	13	14	15
Lena	19.9108	19.9325	19.9459	19.9761	∞
Baboon	20.2850	20.3064	20.4220	20.3743	∞
Peppers	20.0777	20.1004	20.1957	20.1538	∞

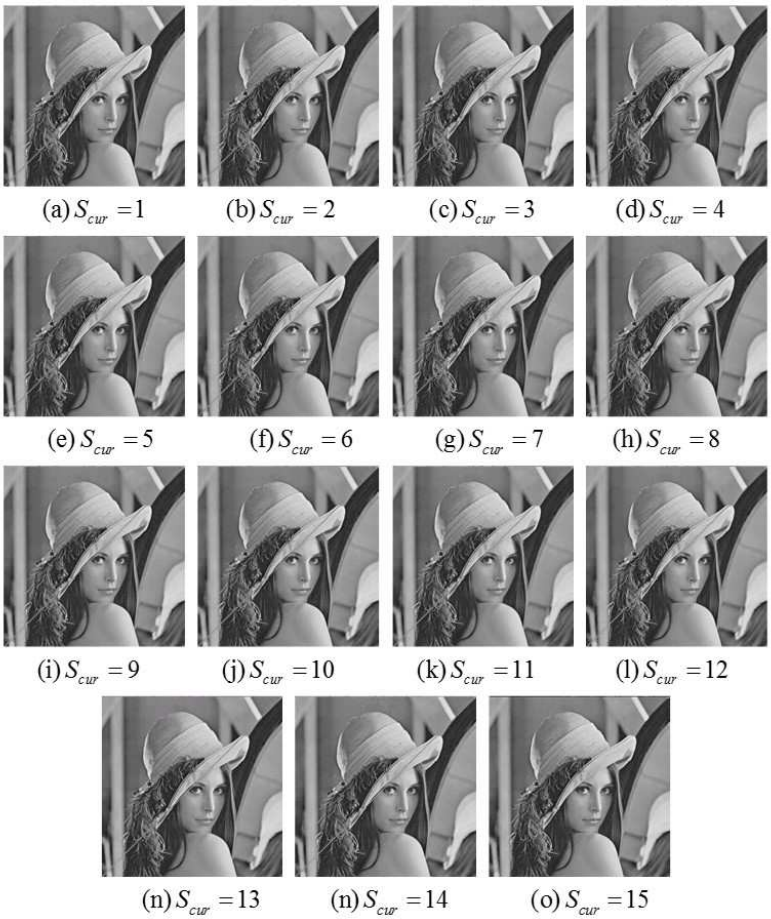


FIGURE 24. (a)-(o) denote the reconstructed images of "Lena" for each round with $S_{total} = 15$ by the proposed scheme

TABLE 4. Comparison of methods, where $r_1, r_2, \ldots, r_t$ denote the t threshold values with $r_1 \leq r_2 \leq \cdots \leq r_t = k$ in the method of Chen et al. , give an example, $r_1 = 2, r_2 = 3, r_3 = 4$. Therefore, the ratio is $3/(2+3+4) = 1/3$

Schemes	Threshold	Size extension ratio	Progressive transmission	Distortion free
Tso[4]	(k,n)	1/k	No	No
Gunzin Ultutas et al. [5]	(k,n)	2/k	No	Yes
Fang[6]	(k,n)	4	Yes	No
Chen et al.[11]	(k,n)	$t/(r_1 + r_2 +, \ldots, +r_t)$	Yes	Yes
Chao et al.[13]	(n,n)	1	Yes	Yes
Proposed scheme	(k,n)	$\frac{1}{k} + \frac{T}{4\times4}$	Yes	Yes

scheme can assist receivers in obtaining secret images with less image quality by using the received data at the first two or three rounds and, then, obtain the distortion free secret image once they receive all transmitted data.

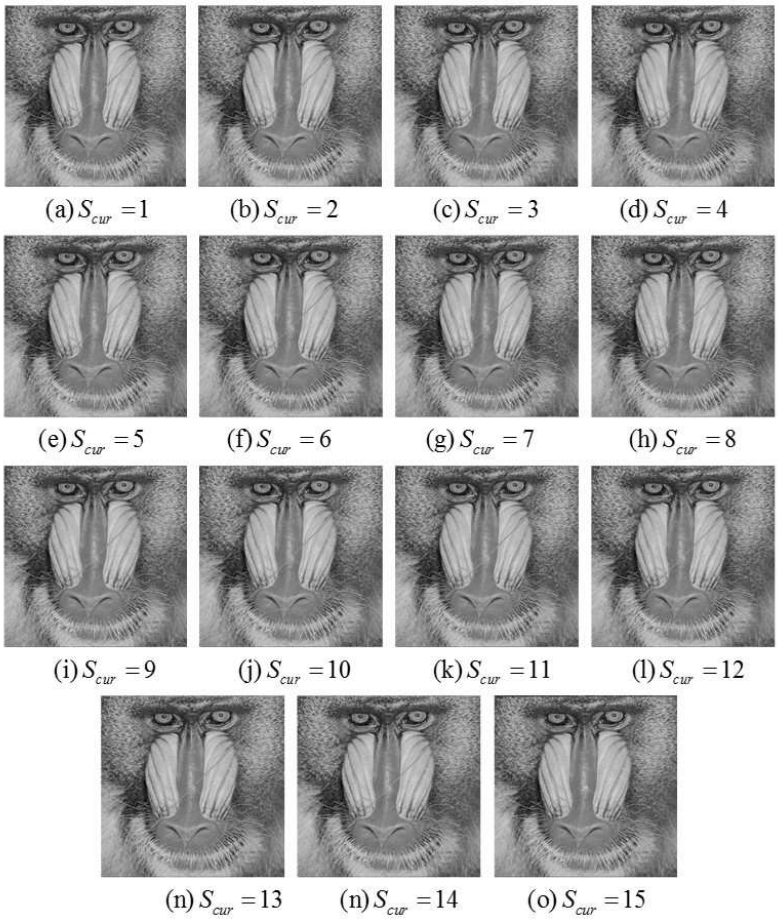


FIGURE 25. (a)-(o) denote the reconstructed images of "Baboon" for each round with $S_{total} = 15$ by the proposed scheme

REFERENCES

[1] Shamir, A., "How to share a secret," *Communications of ACM*, vol. 22, 1979, pp. 612-613.
[2] Blakley, G.R., "Safeguarding vryptographic keys," *Proceedings of the National Computer Conference*, 1979, pp. 313-317.
[3] Thien, C.-C., Lin, J.-C., "Secret image dharing," *Computers and Graphics*, vol. 26, 2002, pp. 765-770.
[4] Tso, H.-K., "Sharing secret images using Blakley's concept," *Optical Engineering*, vol. 47, 2008, pp. 7.
[5] Guzin Ulutas, Mustafa Ulutas, Vasif Nabiyev, "Distortion free geometry based secret image sharing," *Procedia Computer Science*, vol. 3, 2011, pp. 721-726.
[6] Fang, W.-P., "Friendly progressive visual secret sharing," *Pattern Recognition*, vol. 41, no. 4, 2008, pp. 1410-1414.
[7] Huang ,C.-P., Hsieh, C.-H., Huang, P.-S., "Progressive sharing for a secret image," *Journal of Systems and Software*, vol. 83, no. 3, 2010, pp. 517-527.
[8] Chao,C.-W., Hsieh,C.-H., Lu,P.-C., Cheng,T.-A., "Modified block truncation coding for image compression," *Pattern Recognition Letters*, vol. 17, no. 14, 1996, pp. 1499-1506.
[9] Sren I Olsen, "Block truncation and planar image coding," *Pattern Recognition Letters*, vol. 21, no. s 13-14, 2000, pp. 1141-1148.
[10] Paul Salama, Martha Saenz, Edward J. Delp, "Block Truncation Coding*BTC*," *Handbook of Image and Video Processing SecondEdition*, 2005, pp. 661-671.
[11] Chen,S.-K., Lin,J.-C., "Fault-tolerant and progressive transmission of images," *Parallel Computing*, vol. 27, no. 10, 2001, pp. 1381-1399.
[12] Wang, R.-Z., Chien,Y.-F., Lin Y.-Y., "Scalable user-friendly image sharing," *Journal of Visual Communication and Image Representation*, vol. 21, no. 7, 2010, pp. 751-761.

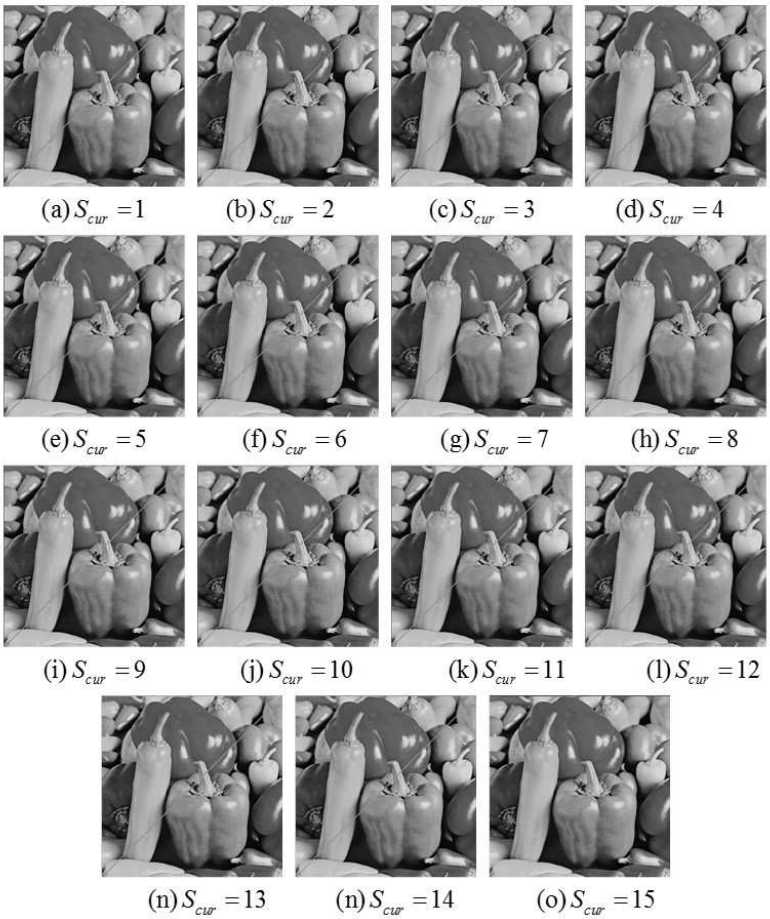


FIGURE 26. (a)-(o) denote the reconstructed images of "Peppers" for each round with $S_{total} = 15$ by the proposed scheme

[13] Chao ,K.-Y., Lin, J.-C., "User-friendly sharing of images: progressive approach based on modulus operations," *Journal of Electronic Imaging*, 18, 2009, 033008.

[14] Chung, K.-L., Tseng, S.-Y., "New progressive image transmission based on quadtree and sharing approach with resolution control," *Pattern Recognition Letters*, vol. 22, no. 14, 2001, pp. 1545-1555.

[15] Benazza-Benyahia, J.-C. Pesquet, "A unifying framework for lossless and progressive image coding," *Pattern Recognition*, vol. 35, no. 3, 2002, pp. 627-638.

[16] Chang, C.-C., Jau,J.-C., Chen,T.S., "A fast reconstruction method for transmitting image progressively," *Consumer Electronics, IEEE Transactions on*, vol. 44 , no. 4, 1998, pp. 1225 - 1233.

[17] Raphael C.-W. Phan, "Reducing the exhaustive key search of the Data Encryption Standard (DES)," *Computer Standards & Interfaces*, vol. 29, no. 5, 2007, pp. 528-530.

[18] B. Burnham, "DES (Data Encryption Standard) cryptographic services designed for the DOE wide band communications network," *Computers & Security*, vol. 7, no. 5, 1988, Page 510.

[19] Santanu Sarkar, Subhamoy Maitra, "Cryptanalysis of RSA with two decryption exponents," *Information Processing Letters*, vol. 110, no. 5, 2010, pp. 178-181.

[20] Lin, I.-C., Chang,C.-C., "Security enhancement for digital signature schemes with fault tolerance in RSA," *Information Sciences*, vol. 177, no. 19, 2007, pp. 4031-4039.

[21] R.M. Davis, "The data encryption standard in perspective," *IEEE Communications Magazine*, vol. 16, no. 6, 1978, pp. 5-9.

[22] W. Dife, "The rst ten years of public-key cryptography," *Proceedings of the IEEE*, vol. 76, no. 5, 1988, pp. 560-577.