HIERARCHICAL SECRET IMAGE SHARING METHOD USING JPEG 2000 CODESTREAM SYNTAX

[1]Masayuki Hashimoto, [2]Yoriyuki Minami, [1]Kenji Matsuo, and [1]Atsushi Koike

[1]Visual Communication Laboratory, KDDI R&D Laboratories Inc.
2-1-15 Ohara, Kamifukuoka-Shi, Saitama, 356-8502, Japan
Phone: + 81 49 278 7851, Fax: + 81 49 2787439, email: masayuki@kddilabs.jp
[2]Department of Electrical Engineering, Faculty of Engineering, Tokyo University of Science
1-3 Kagurazaka, Shinjuku-Ku, Tokyo, 162-8601, Japan

ABSTRACT

This paper proposes a hierarchical secret sharing method for image data. In the conventional secret sharing scheme (SSS), secret data is reconstructed only if more shares are gathered and combine than a pre-decided number, k. Though large k makes a strong security feature, it creates problems with the handling of the many shares required to reconstruct the original data. If it allows us to browse the contents of secret image data roughly, then it relieves us from the troublesome management of the secret data. Therefore, in order to improve usability, we propose a new SSS using JPEG 2000's hierarchical image representation. This method can control the number of reconstructed frequency sub-bands of image information by the number of combined shares. It can also control the precision of pixel values and the reconstructed region in the image. This paper shows the feasibility and efficiency of the proposed method by computer simulations.

1. INTRODUCTION

Because of the popularization of broadband networks, digital contents distribution continues to attract increased attention. When considering the distribution of copyrighted contents, we need solid content-level security technology. The secret sharing scheme (SSS) is one of the solutions for the solid protection of secret information and its secure presentation. A (k, n)-threshold method [1] is a well-known type of SSS. It divides secret information, into n secret share information using a random (k-1)-degree polynomial. At the very least, k shares are required to reconstruct the secret information. Even if at worst (k-1) shares are leaked out, the secret information will not be leaked. Furthermore, even if (n-k) shares are lost at worst, the secret information can be reconstructed by collecting the other shares. This means strong content-level security. However, even a summary of secret information cannot be browsed until k shares are completely collected by the (k, n)-threshold method. Though large k is a strong security feature, it causes problems with the handling of the many shares required to reconstruct the original data. Even if the image quality is degraded from the original quality or a particular region of the image is masked, the image can be used for some applications, for example, image browsing. Usability of the (k, n)-threshold method is expected to be improved for image data if a smaller number of shares can generate a lower quality image or a partially masked image than k which is the number of shares required to reconstruct original information.

Therefore, we propose a hierarchical SSS which can control the reconstructed image quality and reconstructed region of the image by the number of combined shares. In order to realize the proposed method, we use the SSS and JPEG 2000's[2] hierarchical representation of image information. In the JPEG 2000 (JP2) coding method, image information is divided into frequency sub-bands using discrete wavelet transformation (DWT). Furthermore, coded data can be hierarchically divided depending on quantization accuracy. Image information can be divided spatially into several regions called "tiles," which allows random access to a particular region of an image. Code bits which belong to a particular tile, the decomposition level of DWT and a layer are packed into a packet which is a unit data segment of JPEG 2000. Resolution scalability and image quality scalability are provided by selecting packets which are sent to a JPEG 2000 decoder.

The proposed method controls the value of k for each JPEG 2000 packet when conducting SSS. This allows the number of packets which are correctly reconstructed to be changed depending on the number of combined shares. In other words, a small number of the combined shares means reconstruction of only the packets generated with smaller k than the number of the combined shares. Contrarily, the larger number of the combined shares means reconstruction of a larger number of packets. Generally, decoded JPEG 2000 image quality can be controlled by which packets are being decoded. Therefore, using an appropriate value of k for the sharing process of each packet allows us to control the quality of a reconstructed image by the number of combined shares.

In this paper, firstly, we discuss SSS in Sect. 2. Next, the JPEG 2000 coding algorithm and code-stream syntax are described in Sect. 3. Then, the proposed method is explained in Sect. 4. Sect. 5 presents experiments for evaluating the feasibility and efficiency of the proposed method. Finally, we conclude this paper in Sect. 6.

2. SECRET SHARING METHODS

The distributed storage method using SSS is a method in which data is divided into serial shares and the shares are distributively stored. Then original information can be reconstructed by combining some of these shares. The distributed storage method with SSS has strong security features: If some of the shares are lost by damage of a storage device, original data can be reconstructed by the other shares. Furthermore, even if a part of the shares is leaked out, original secret information is not leaked as long as the number of the leaked shares is small.

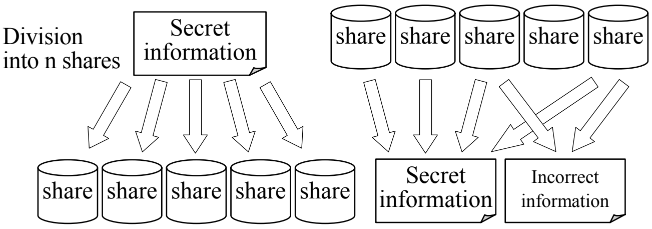


Fig. 1 Secret Sharing Scheme

A well-known SSS is the (k, n)-threshold secret sharing scheme. It divides secret information into n secret share information, W_i ($1 \leq i \leq n$) using a random (k-1)-degree polynomial over the finite Galois Field GF(p).

$$f(x) = S + r_1 x + \cdots + r_{k-1} x^{k-1} (\text{mod } p) \quad (1)$$

At the very least k shares are required to reconstruct S, where k ≤ n. A perfect threshold scheme is a threshold scheme in which knowledge of (k-1) or fewer shares does not provide any advantage to the opponent to find the secret. The following tasks are performed to generate shares.

1. Choose prime p larger than n and the secret S.
2. Construct f(s) by selecting (k-1) random coefficients r_1 to r_{k-1}
3. Compute the shares W_i by evaluating f(x) at n distinct points.

The secret S can be recovered by constructing the polynomial from any k of the n shares.

$$S = f(0) = \sum_{i=1}^{k} c_j W_i, which\ c_j = \prod_{\substack{j=1 \\ j \neq i}}^{k} \frac{x_j}{x_j - x_i} \quad (2)$$

3. HIERARCHICAL IMAGE CODING WITH JPEG 2000

3.1 Coding Algorithm of JEPG 2000

JPEG 2000 is one method of representing image information hierarchically. The processing flow of the JPEG 2000 encoder is shown in Fig. 2. First, image is divided into one or more tiles, which are rectangular pixel regions. This allows easy random access to a particular region of the encoded image. Next, tiles are analyzed into four frequency sub-bands using discrete wavelet transformation (DWT). The lowest frequency sub-band is analyzed in recursive fashion. The decomposition level of a sub-band is defined as the number of repetitions of 2-dimension DWT generating the sub-band. Then DWT coefficients are quantized if necessary.

The sub-band is divided into several code-blocks which are coding units of JPEG 2000. Bit-plane based coding is conducted for DWT coefficients in code-blocks. Figure 3 shows the bit-plane coding process. Basically, a bit-plane is scanned three times creating three coding passes. These are the significant propagation pass, which is shown as "a" in Fig. 3, the magnitude refinement pass, which is shown as "b" in the figure, and the clean up pass, which is shown as "c." Each bit of the bit-plane is scanned once by one of the three passes.

Coding passes are encoded by an arithmetic coder using context information which shows the status of bits located around the bits that are being coded. Coded passes in a code-block are allocated to one or more layers in order to realize scalability of the signal-to-noise ratio (SNR), which reflects image quality. Coding passes are hierarchically allocated to layers according to their contribution to image quality. Compressed data of coding passes allocated in the same layer have almost the same degree of contribution to image quality.

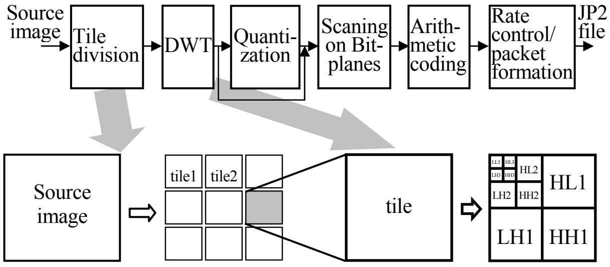


Fig. 2 Processing flow of JPEG 2000

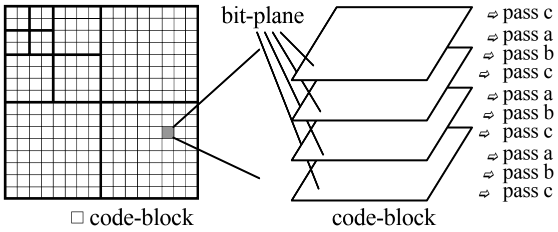


Fig. 3 Code-block and pass in JPEG 2000 coding algorithm

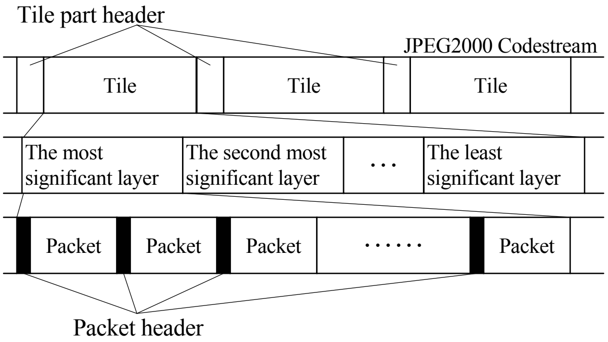


Fig. 4(a) Codestream syntax of JPEG 2000 (layer progressive mode)

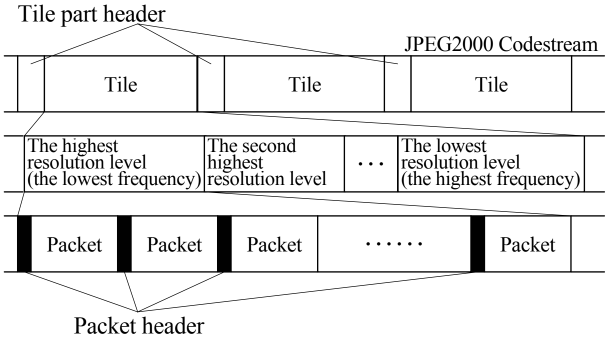

Tile part header
JPEG2000 Codestream
Tile
Tile
Tile
The highest resolution level (the lowest frequency)
The second highest resolution level
. . .
The lowest resolution level (the highest frequency)
Packet
Packet
Packet
. . . . . .
Packet
Packet header

Fig. 4(b) Codestream syntax of JPEG 2000 (resolution progressive mode)

3.2 Code-Stream Syntax of JPEG 2000: Tile and Packet

Basically, all compressed bits for a tile are allocated in serial order. Code bits which belong to a particular tile, the decomposition level of DWT and the layer are packed into a packet which is a unit data segment of the JPEG 2000 code-stream syntax. Resolution scalability and image quality scalability are provided by selecting packets which are sent to a JPEG 2000 decoder.

In the quality scalability mode, as more packets are decoded, image quality is improved. All packets related to a layer are allocated in a row fashion, and the more significant layer appears more forward in a code-stream. Figure 4(a) shows this type of packet formation. In resolution scalability mode, the first few bytes are used to represent a low frequency component. As more bytes are decoded, the resolution of the image increases by factors of 2 on each side. Eventually, the full size image is obtained. Figure 4(b) shows this type of packet formation.

4. PROPOSED METHOD

4.1 Share Generation Process

Figures 5 and 6 show respectively the processing flow and share generation of the proposed method. First, a secret image is normally encoded in a JPEG 2000 code-stream with the resolution progressive mode or layer progressive mode. Then JPEG 2000 share files are generated based on the original JPEG 2000 code-stream. At the share generation stage, the value of k is decided for each hierarchical component (a layer or a resolution level). This allows the hierarchical component to be correctly reconstructed using k shares. In the proposed method, a smaller value of k is used for the hierarchical component with higher priority.

For instance, hereafter, we consider share generation whereby the number of layers that are correctly reconstructed increases as more shares are combined. If we define k_i as the value of k for the i-th layer, the value of k_i is decided as follows: $k_1 \le k_2 \le \cdots \le k_L$, where L shows the total number of layers. Then, data of the packet in the i-th layer of each share is generated with k_i from the original JPEG 2000 packet data. We define the share's ID which is used as x in equation (1) and (2). The share data generation is conducted by the M bits: M bits information of the original packet data is used as secret information, S, to generate M bits in the share's packet data using the share's ID. In this paper M = 8 is used for computer simulation of the proposed method. Each share's ID is embedded in a comment tag in each JPEG 2000 share file because we are allowed to store private information in a comment tag of JPEG 2000.

Because only packet data have been converted in the generated JPEG 2000 share files, they are completely based on JPEG 2000 code-stream syntax. Thus, they can be decoded by an ordinary JPEG 2000 decoder. However, the decoded image is noisy and completely different from the original image because the packet data is converted by the share generator.

The data size of each JPEG 2000 share file is the same as that of the original JPEG 2000 code-stream. Therefore, if n share files are generated, the total data size is n times larger than the original code-stream size. That data size overhead is the same as that of the classical (k, n)-threshold method, where there is no hierarchical division of image data such as for decomposition levels, layers or tiles.

4.2 Reconstruction Process

The following shows a reconstruction process of the proposed method. Assume that m shares, whose original image is identical, are collected. They have the same header information except for the comment tag where each share's ID is stored. Then, the JPEG 2000 decoding process can be started using the header information of one of them arbitrarily. Before decoding JPEG 2000 packet data, packet data is reconstructed by equation (2) using the share's ID and corresponding packets of all shares. If $k_i \ge$ m, the i-th layer is correctly reconstructed. The other layers are not correctly reconstructed. Then the reconstructed packet data is decoded.

The processes described above decode DWT coefficients. The ordinary decoding process in JPEG 2000 is then conducted and an image is reconstructed. If some layers were not reconstructed correctly at the previous packet reconstruction stage, the decoded image contains some degree of noise. Equally, the proposed method can be used with the JPEG 2000 code-stream having resolution progression.

5. EXPERIMENTAL RESULTS AND EVALUATION

We verify the feasibility and efficiency of the proposed method. The following three types of proposed method are evaluated: The first the method whose base progression mode is resolution level progression, is explained in Sect. 5.1. The second is the type whose base progressive mode is layer progression, which is evaluated in Sect. 5.2. The last is the type in which a particular tile's data is converted by the share generator, which is described in Sect.5. 3.

5.1 Resolution Progressive Mode

In this section, the proposed method based on the resolution progressive mode is evaluated. The encoder creates a base JPEG 2000 code-stream which has seven resolution levels. Here we define k_i as the value of k for the resolution level which contains the i-th lowest frequency component. At the share generator, $(k_1, k_2, \cdots k_7) = (2, 3, \cdots , 8)$ and n = 10 which means the generator creates 10 share files. We define m as the number of share files combined. Figure 7(a) shows the reconstructed image when m = 1. Figures 7(b), (c), (d) and (e) respectively show the reconstructed image when m = 2, 3, 4 and 8. In Fig. 7(a), the reconstructed image from one share does not show image information at all. Figures 7(b) to (e) show that larger m shows less noise on the reconstructed images. This occurs because more resolution levels are reconstructed as more shares are combined. In this case, 8 or more shares completely create the original image.

5.2 Layer Progressive Mode

In this section, the proposed method based on the layer progressive mode is evaluated. The encoder creates the base JPEG 2000 code-stream which has four layers. The allocated bit rate for the first layer is 0.001 bit/pixel (bpp), for the second layer it is 0.004 bpp and for the third layer it is 0.045 bpp. Here, we define k_i as the value of k for the i-th layer. At the share generator, $(k_1, k_2, k_3, k_4) = (2, 3, 4, 5)$ and n = 10. Figure 8(a) shows the reconstructed image when m = 1. Figures 8(b), (c), (d) and (e) respectively show the reconstructed image when m = 2, 3, 4 and 5. In Fig. 8(a), the reconstructed image from one share doesn't show image information at all. Figures 8(b) to (e) show that larger m shows less noise on the reconstructed images. This occurs because more layers are reconstructed as more shares are combined. In this case, 5 or more shares completely create the original image.

5.3 Control of Reconstructed Tile

In this section, the proposed method based on the tiled JPEG 2000 code-stream is evaluated. The encoder creates the base JPEG 2000 code-stream which is divided into 20 tiles shown in a tile layout in Fig. 9. k = 3 is used for tile A in the figure and k = 2 is used for tile B. n is 10 in this experiment. Figure 10(a) shows the reconstructed image when m = 1. Figures 10(b) and (c) respectively show the reconstructed image when m = 2 and 3. Figure 10(a) shows the reconstructed image from one share and does not show image information in both tile A and tile B. Figure 10(b) shows the reconstructed image from 2 shares is masked only in tile A. The image appears in tile B. This occurs because tile A needs 3 shares to appear and tile B needs 2 share to appear. In this case, 3 or more shares completely create the original image.

6. CONCLUSION

This paper proposes a hierarchical secret sharing method for image data in order to improve the usability of SSS. The proposed method can control the number of the reconstructed frequency sub-bands of image information by the number of combined shares. It can also control the precision of pixel values and the reconstructed region in the image. We verified the feasibility and efficiency of the proposed method.

REFERENCES

[1] A. Shamir, "How to share a secret," Communications of the ACM, Vol. 22, No. 11, pp. 612-613, November 1979.

[2] ISO/IEC 15444-1, "Information technology – JPEG 2000 image coding system…," ISO/IEC JTC 1/SC 29/WG1, Jan.2001.

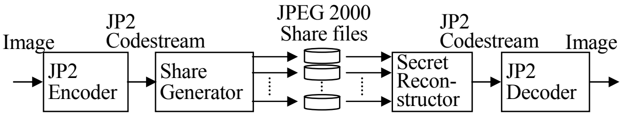


Fig. 5 Flow of the proposed method

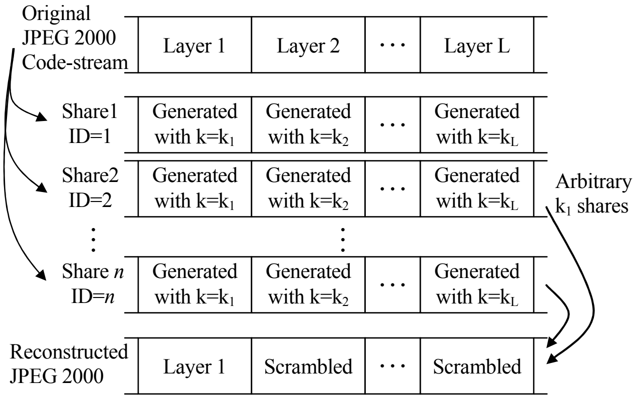
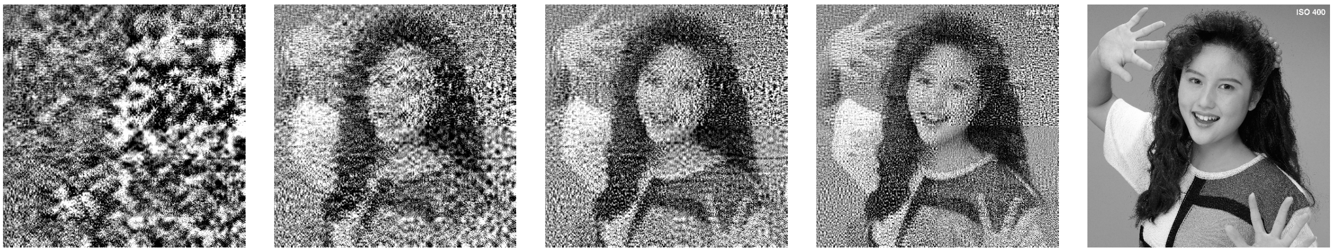


Fig. 6 Share generation and reconstruction



(a) Reconstructed image. m = 1. (PSNR 7.3dB)

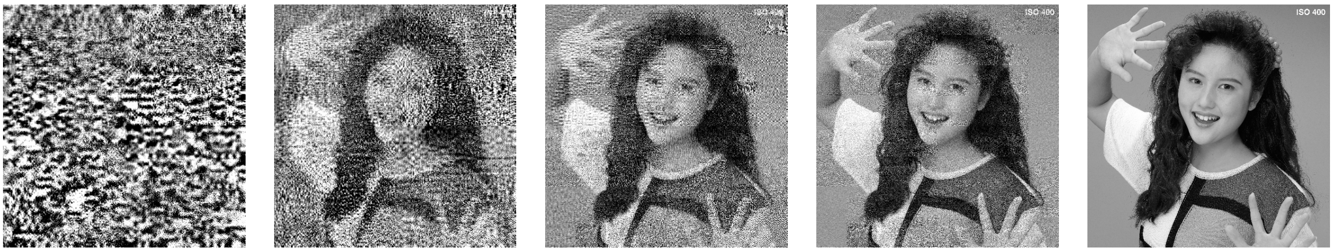
(b) Reconstructed image. m = 2. (PSNR 10.2dB)

(c) Reconstructed image m = 3. (PSNR 10.7dB)

(c) Reconstructed image. m = 4. (PSNR 11.3dB)

(d) Reconstructed image. m = 8.(Lossless)

Fig. 7 Reconstructed images of the proposed method with resolution level progression



(a) Reconstructed image. m = 1. (PSNR 6.8dB)

(b) Reconstructed image. m = 2. (PSNR 10.4dB)

(c) Reconstructed image m = 3. (PSNR 12.3dB)

(c) Reconstructed image. m = 4. (PSNR 15.6dB)

(d) Reconstructed image. m = 5.(Lossless)

Fig. 8 Reconstructed images of the proposed method with layer progression

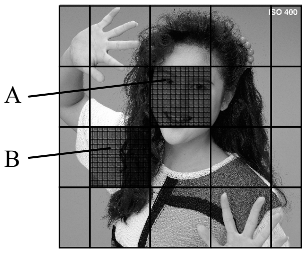
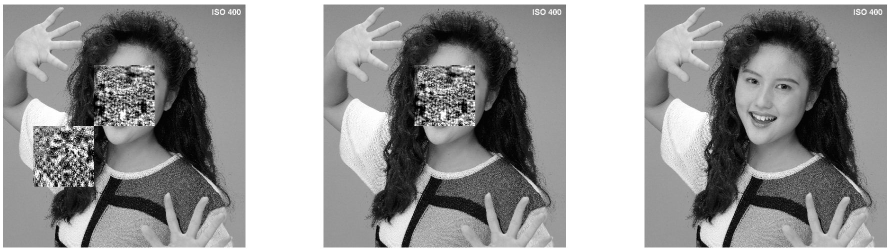


Fig. 9 Tile layout: Data of tile A and tile B are converted.



(a) Reconstructed image. m = 1. (PSNR 16.4dB)

(b) Reconstructed image. m = 2. (PSNR 21.3dB)

(c) Reconstructed image. m = 3.(Lossless)

Fig. 10 Reconstructed image of the proposed method controlling masked tiles