BPCS steganography using EZW lossy compressed images

Jeremiah Spaulding †, Hideki Noda *†, Mahdad N. Shirazi ‡, Eiji Kawaguchi †

†Kyushu Institute of Technology,
Dept. of Electrical, Electronic & Computer Engineering,
1-1 Sensui-cho, Tobata-ku, Kitakyushu, 804-8550 Japan

‡Communications Research Laboratory,
Keihanna Human Info-Communications Research Center,
2-2-2 Hikaridai, Seika-cho, Soraku-gun, Kyoto, 619-0289 Japan

Abstract

This paper presents a steganography method based on an embedded zerotree wavelet (EZW) compression scheme and bit-plane complexity segmentation (BPCS) steganography. The proposed steganography enables us to use lossy compressed images as dummy files in bit-plane-based steganographic algorithms. Large embedding rates of around 25% of the compressed image size were achieved with little noticeable degradation in image quality.

keywords: data hiding, data security, steganography, wavelet, image compression

1 Introduction

With the Internet boom of the last ten years and the even more recent boom in e-commerce, data security has become a very important field of study. There are many good security protocols in use, such as public key encryption and SSL. These overt methods cause the data, if intercepted by an unauthorized party, to be unintelligible and useless. However, as it has been proven again and again in the past, today's best encryption may be broken tomorrow by the constant advancement of the art. The topic of this paper is a

*Corresponding author: Tel: +81-93-884-3247, Fax: +81-93-871-5835, Email: noda@know.comp.kyutech.ac.jp

form of security through obscurity. If nobody knows the encrypted data is there, then they cannot try to break its protection. Steganography is the practise of hiding or camouflaging secret data in an innocent looking dummy container. This container may be a digital still image, audio file, movie file, or even a printed image. Once the data has been embedded, it may be transferred across insecure lines or posted in public places. Therefore, the dummy container should seem innocent under most examinations.

In previous steganographic algorithms, bit-plane decomposition[1] was commonly used (Katzenbeisser and Petitcolas, 2000; Wang et al., 2001) combined with the simple approach of replacing the binary data in the least significant bit-planes with secret binary data (Katzenbeisser and Petitcolas, 2000). Previous work presented a sophisticated steganography method, called bit-plane complexity segmentation (BPCS) steganography, which makes use of bit-plane decomposition and the characteristics of the human vision system (Niimi et al., 1998). Noting that the human eye cannot perceive any shape information in a very complicated binary pattern, we can replace noise-like regions in the bit-planes of the dummy image with secret data without deterioration of image quality. BPCS steganography has proven to be very effective in embedding data into many classes of dummy files including 8-bit gray images (Niimi et al., 1998), 24-bit true color images (Kawaguchi and Eason, 1998) and 8-bit indexed color images (Ouellette et al., 2000). BPCS steganography has also been successfully applied to stereo and mono digital audio files (Kusatsu et al., 1998). The benefits of this technique over traditional steganography are the very large percentage (30%-50%) of the dummy file that can be replaced with secret data and the lower occurrence of visual artifacts in the post-embedding image.

There have been very few steganographic techniques which consider compression of dummy files, i.e., steganography applicable to lossy compressed images (Kataoka et al., 1989; Kobayashi et al., 2000; Chung et al., 2001). Kataoka et al. (1989) presented a data embedding method into adaptive discrete cosine transform (ADCT) coded images and Kobayashi et al. (2000) presented a data embedding method into JPEG bitstreams. However the embedding capacity ((embedded data size)/(compressed image file size)) in these methods is very limited: around 6% for the former method and around 2% for the latter method. Chung et al. (2001) presented a singular value decomposition (SVD)- and vector quantization (VQ)-based image hiding scheme. However it is effective only for hiding one image's data in a dummy image. For BPCS steganography as well as other bit-plane-based steganography methods, one has to face the problem that lossy compression of an embedded dummy image can easily lead to destruction of the

[1] For example, an n-bit image can be decomposed into a set of n binary images by bit-slicing operations (Jain, 1989).

embedded information. This is a critical problem since most media files are stored and transmitted in lossy compressed formats.

This paper presents a steganography method based on a lossy wavelet compression scheme and BPCS steganography. This algorithm utilizes the embedded zerotree wavelet (EZW) compression scheme proposed by Shapiro (1993), where the wavelet coefficients of an image are quantized into a bit-plane structure. The proposed method provides a seamless integration of the EZW lossy compression scheme and the BPCS steganography method and a solution to the aforementioned problem associated with bit-plane-based steganography methods. Note that lossy compressed images become available as dummy data in the proposed method but another lossy compression applied to an already embedded dummy image can easily destroy the embedded information. The proposed method can be applied to other wavelet-based lossy compression schemes like SPIHT (Said and Pearlman, 1996) and JPEG2000, because in these compression schemes the wavelet coefficients are also quantized into a bit-plane structure.

The rest of this paper is organized as follows. In Section 2, necessary background on BPCS steganography is given, followed by information on wavelet compression and the EZW algorithm in Section 3. In Section 4, BPCS steganography using EZW compressed images is presented. The paper continues with experiments done with an implementation of the proposed algorithm in Section 5. Conclusions from the experiments are addressed in Section 6.

2 BPCS steganography

BPCS steganography addresses the embedding limit by working to disguise the visual artifacts that are produced by the steganographic process. Optometric studies have shown that the human vision system is very good at spotting anomalies in areas of homogenous color, but less adept at seeing them in visually complex areas. When an image is decomposed into bit-planes, the complexity of each region can be measured. Areas of low complexity such as homogenous color or simple shapes appear as uniform areas with very few changes between one and zero. Complex areas such as a picture of a forest would appear as noise-like regions with many changes between one and zero. These random-seeming regions in each bit-plane can then be replaced with hidden data, which is ideally also noise-like. Because it is difficult for the human eye to distinguish differences between the two noise-like areas, we are able to disguise the changes to the image. Additionally, since complex areas of an image tend to be complex through many of

their bit-planes, much more data can be embedded with this technique than with those that are limited to only the lowest planes.

In BPCS steganography, the complexity of each subsection of a bit-plane is defined as the number of non-edge transitions from 1 to 0 and 0 to 1, both horizontally and vertically. For any square of $n \times n$ pixels, the maximum complexity is $2n(n-1)$ and the minimum is of course 0. In Fig. 1, white represents a one and black a zero. Both squares, or patches, have the same number of ones and zeros, but very different complexities. This shows that one contains much more visual information than the other. The complex patch (a) has very little visually informative information, therefore it can be replaced with secret data and have a very small effect on the image's quality. However, if the more visually informative patch (b) was replaced, it would cause noise-like distortion of the definite edges and shapes.

A typical procedure for data hiding in BPCS steganography is summarized as follows.

(1) Segment each bit-plane of a dummy image into 8×8 size blocks. Then classify these blocks into informative and noise-like blocks using a threshold of the complexity α_0. A typical value of α_0 is $0.3\alpha_{max}$, where α_{max} is the maximum possible complexity value.

(2) Segment a secret file into a series of blocks each containing 8 bytes of data. These blocks (which we call secret blocks) are regarded as 8×8 binary images.

(3) If a secret block is less complex than the threshold α_0, conjugate it to make it more complex. Here the process called conjugation is the exclusive OR operation with a checkerboard pattern. The relation $\alpha^* = \alpha_{max} - \alpha$ holds true (Niimi et al., 1998), where α and α^* are the complexity of a given image and that of the conjugated image, respectively.

(4) Replace each noise-like block in the bit-planes with a block of secret data. If the block is conjugated, then record this fact in a conjugation map.

(5) Also embed the conjugation map in the same way as the secret blocks.

The decoding procedure to extract the embedded secret data is just the reverse of the embedding procedure. In the decoding process, the embedding threshold α_0 and amount of secret data need to be known. The amount of secret data can be embedded into a specific place in the dummy file.

This technique works very well with natural images, as they tend to have many areas of high complexity. Images with many complex textures and well shaded objects usually have a high embedded data capacity.

BPCS steganography works much less well with computer generated images and line art, as those classes of images tend to have large areas of uniformity and sharply defined border areas. With these types of images, there is very little complexity to exploit and any changes tend to generate very obvious artifacts. This is one flaw BPCS steganography shares with traditional steganography. Another shared flaw is the fragility of the secret data with respect to changes in the post-embedding image. Any lossy compression will corrupt the hidden data, as will most transformations and filters. Since this makes the hidden data very vulnerable to any destructive attack, BPCS steganography is almost useless for watermarking purposes. However, depending on the desired application, this property may actually be a benefit. In the realm of data hiding, it can be a good thing, allowing for easy destruction of the secret message after completing the extraction of the embedded message. This allows retention of the dummy image, which is now completely innocent.

3 Wavelet compression and EZW encoder

The original JPEG standard made use of the block discrete cosine transform (DCT) for its compression. This standard has been in wide use, having gained popularity in part because of the demand for a good standard compression scheme to speed the download of images from the newly popular World Wide Web. At the same time, wavelet image compression using the discrete wavelet transform (DWT) was in the early stages of research and beginning to gain acceptance in the academic community. After being refined, the DWT-based techniques achieved even better compression than the DCT-based ones, with fewer artifacts and distortions. There have been great advances in the field of wavelet compression within recent years and many of today's best image, audio, and video COmpressor/DEcompressors (CODECs) are based on wavelets, including JPEG2000.

Shapiro (1993) presented a seminal paper on a wavelet image compression algorithm called embedded zerotree wavelet (EZW) compression. The EZW encoder is simple, fast and provides very good compression rates. It takes advantage of the correlations between subbands in a wavelet coefficient set to lower the amount of information needed to represent them. The successive approximation method used by the EZW algorithm encodes the wavelet coefficients one bit-plane at a time, starting with the most significant bit. In EZW compression, each wavelet coefficient w is expressed as

$$w = T(a_0 + a_1 2^{-1} + a_2 2^{-2} + \cdots), \quad a_i \in \{0, 1\}, \tag{1}$$

where T is a constant satisfying $T > 0.5w_{max}$ (w_{max} is the maximum absolute value among all wavelet coefficients in a DWT image). Typically $T = 2^{\lfloor \log_2 w_{max} \rfloor}$ is used. Since $(a_0 + a_1 2^{-1} + a_2 2^{-2} + \cdots)$ is a binary expression, the DWT image can be considered to be a bit-plane structure. EZW encoding is conducted from higher bit-planes to lower ones. That is to say, encoding starts with more important information, so that decoding can be performed on the most important information first. Therefore, even if decoding is discontinued before the end of the image file, almost optimal decoding results can still be achieved under the reduced amount of information. Thus, this type of compression, which is called progressive compression, is particularly suitable for Internet communication.

Other wavelet-based compression schemes are also progressive, where the wavelet coefficients are represented in similar ways as in EZW. Those include SPIHT (Said and Pearlman, 1996) and JPEG2000 for still images, 3-D SPIHT (Kim et al., 2000) and motion-JPEG2000 for video, and a wavelet-packet-based compression method (Srinivasan and Jamieson, 1998) for audio data.

4 BPCS steganography combined with EZW compression

This paper proposes a method of embedding secret data into a DWT-based lossy compressed image using the previously described BPCS steganography. The coefficients of the DWT have many image-like properties, and BPCS steganography is ideal for exploiting them. The main properties leveraged for BPCS steganography are:

· Correspondence: Spatial areas in each section of the coefficient subbands correspond directly to areas in the original image.

· Complexity: The bit-planes at corresponding significance levels of the wavelet coefficients and the original image are usually proportionally complex.

· Resilience: Changes in the values of the wavelet coefficients do not create disproportionately large changes in the reconstructed image.

The property of correspondence states that in each subband of the wavelet coefficients, any subsection of that subband directly corresponds to a section of the original image. This is a scaled relationship, as the subbands decrease in size by a factor of two with each iteration of the multi-scale DWT. For example, an 8×8 patch of pixels in the original image corresponds to a 4×4 patch of pixels in the finest (first scale) subband. This allows the same complexity metrics to be used on the wavelet coefficients as are used on

the original image.

In the wavelet coefficients, the complexity of any subsection is related to the complexity of the corresponding subsection of the original image. While the amount of complexity in the wavelet coefficients is very important, the distribution of the complexity is also important. In the wavelet coefficients, the bits are ordered in decreasing significance, just as in the original image. Because of this, bit-planes tend to become more complex towards the least significant bits. This is good for BPCS steganography because this is where changes will have the smallest impact.

The capacity of a container image is limited not only by its complexity, but by the decoder's resilience to changes made in the coefficients. Resilience indicates the ability of the wavelet coefficients to absorb changes in value without changing the final image. The more resilient they are, the more changes that can be made and thus the more data that can be embedded. The inverse DWT (IDWT) is quite resilient to small changes in the coefficient values, and large changes experience a blending and blurring effect from the smoothing nature of the wavelet transform. This property is extremely useful for BPCS steganography, as many slight changes in the coefficients are blended out and result in very little visual impact on the reconstructed image.

The procedure for data embedding with EZW compression is shown in Fig. 2. The EZW compression algorithm consists of three steps; first, the DWT is applied to an original image, then the EZW encoder performs quantization of wavelet coefficients and outputs a symbol stream, and finally the arithmetic encoder is applied to it and a bit stream (compressed image file) is produced. Data embedding is carried out after the second step, the EZW encoder, and before the third step, the arithmetic encoder. The data embedding procedure is shown by the dashed block in Fig. 2. Applying the EZW decoder to the symbol stream produces quantized wavelet coefficients[2] . Using these quantized wavelet coefficients, bit-planes for the wavelet coefficients can be constructed and used to embed secret data with BPCS steganography. The quantized wavelet coefficients modified by embedding are then subjected to EZW encoding to produce a symbol stream, which is then passed to the arithmetic encoder. Data embedding in an already compressed image file is also possible. A compressed image file, i.e., a bit stream is subjected to arithmetic decoding, and the derived symbol stream is passed to the EZW decoder (see dashed arrows in top-right portion of

[2] In principle the two steps of EZW encoding and EZW decoding are unnecessary to obtain the quantized wavelet coefficients. However, the two steps are performed so that the bit stream may be truncated to meet pre-embedding compression rate requirements.

Fig. 2). Then the following procedure is the same as the aforementioned one shown in the dashed block in Fig. 2.

The data extraction procedure for the encoded image is shown in Fig. 3. EZW reconstruction also consists of three steps; first, the arithmetic decoder is applied to a bit stream, then the derived symbol stream is passed to the EZW decoder to derive quantized wavelet coefficients, and finally the IDWT is applied to them to produce the reconstructed image. Extraction of secret data is carried out by the BPCS method using the quantized wavelet coefficients. That is, the BPCS method can extract secret data from the bit-planes which are constructed during the second step. Generally the data extraction starts after the entire file of the bit stream has been received. Error detection measures, such as embedding the amount of secret data as described in Section 2, can be used to check whether the bit stream has been truncated in transit.

5 Experimental results

The process described in the previous sections was implemented and tested on several standard images including "Lena" and "Barbara". These images are 8 bits per pixel (bpp) gray images, which were 512×512 pixels in size. A 5-scale wavelet transform with the Daubechies 9/7 filter was applied to images. The Moffat (Moffat et al., 1998) adaptive arithmetic encoder was used. The number of bit-planes in the EZW compression was set to 8 and 9.

A subband and bit-plane based weighting scheme was devised to increase the complexity threshold for embedding in the more significant subbands and in the more significant bit-planes. This was necessary because the coefficients in the more significant subbands are less resilient, and changes in the higher bit-planes can cause greater distortion. This decreased the embedding potential for each bit-plane, but resulted in vastly reduced visual distortion. The following formula was used to increase the complexity threshold, which was experimentally set.

$$\alpha = \alpha_0 + \{\frac{2}{3}(1 - \frac{n}{n_{max}}) + \frac{1}{3}(\frac{s-1}{s_{max}})\}\alpha_{max}, \tag{2}$$

where α is the complexity threshold for the s-th scale subband (scale numbered from the finest scale) in the n-th significant bit-plane, α_0 is the base complexity threshold for the first (finest) scale subband in the least significant bit-plane, n_{max} is the total number of bit-planes, s_{max} is the total number of scales in wavelet transform, and α_{max} is the maximum possible complexity value. Those values of α_0, n_{max} and

the number of planes used for embedding are shown in Table 1 and Table 2. $s_{max} = 5$ since the 5-scale wavelet transform was used, and $\alpha_{max} = 24$ since the 4×4 patch size was used as an embedding unit.

Experimental results are shown in Fig. 4 and Table 1 for "Lena", and Fig. 5 and Table 2 for "Barbara". Those images in Figs. 4 and 5 are closeup images. Here random binary data was used as secret data. In Figs. 4 and 5, and Tables 1 and 2, (a)s are for the EZW compressed images using 8 bit-planes, (b)s are for embedding results using (a)s with almost no degradation in image quality, (c)s are for those with little noticeable degradation, and (d)s are for those with obvious distortions. Similarly, (e)s, (f)s, (g)s, and (h)s are relevant results, using 9 bit-planes in the EZW compression. Generally, the proposed steganography was able to achieve embedding rates of around 25% of the final compressed image size with little or no noticeable degradation in image quality. Rates of over 40% were attained but some distortion could be seen.

6 Conclusions

In this paper, BPCS steganography combined with EZW compression was presented, which enables us to use lossy compressed images as dummy files in bit-plane-based steganography methods. However, note that another lossy compression applied to an already embedded dummy image can easily destroy the embedded information. Large embedding rates of around 25% of the compressed image size were achieved with little noticeable degradation in image quality.

The proposed method can be applied to other wavelet-based progressive compression schemes like SPIHT and JPEG2000, because in these compression schemes as well as EZW compression, the wavelet coefficients are quantized into a bit-plane structure. The proposed method is currently being implemented for JPEG2000 compression (Part 1 of JPEG2000 for still image compression). This will allow many more people access to the benefits of BPCS steganography. Application of the proposed method to progressively compressed video is another important issue for future work.

Acknowledgement This work was partly supported by the Telecommunications Advancement Foundation, Japan.

References

Chung, K.L., Shen, C.H., Chang, L.C., 2001. A novel SVD- and VQ-based image hiding scheme. Pattern Recognition Letters 22, 1051-1058.

Jain, A.K., 1989. Fundamentals of Digital Image Processing. Prentice Hall.

Kataoka, T., Tanaka, K., Nakamura, Y., Matsui, K., 1989. Embedding a document into color picture data under adaptive discrete cosine transform coding. Trans. of IEICE J72-B-I, 1210-1216.

Katzenbeisser, S., Petitcolas F.A.P., 2000. Information Hiding Techniques for Steganography and Digital Watermarking. Artech House.

Kawaguchi, E., Eason, R.O., 1998. Principle and applications of BPCS-steganography. Proc. of SPIE 3528, 464-473.

Kim, B.J., Zixiang, X., Pearlman, W.A., 2000. Low bit-rate scalable video coding with 3-D set partitioning in hierarchical trees (3-D SPIHT). IEEE Trans. Circuits and Systems for Video Technology 10, 1374-1387.

Kobayashi, H., Noguchi, Y., Kiya, H., 2000. A method of embedding binary data into JPEG bitstreams. Trans. of IEICE J83-D-II, 1469-1476.

Kusatsu, I., Niimi, M., Noda, H., Kawaguchi, E., 1998. A large capacity steganography using acoustic dummy data. Technical Report of IEICE EA98-69-78, 27-32.

Moffat, A., Neal, R., Witten, I.H., 1998. Arithmetic coding revisited. ACM Trans. Information Systems 16, 256-294.

Niimi, M., Noda, H., Kawaguchi, E., 1998. A steganography based on region segmentation by using complexity measure. Trans. of IEICE J81-D-II, 1132-1140.

Ouellette, R., Noda, H., Niimi, M., Kawaguchi, E., 2000. Topological ordered color table for BPCS steganography using indexed color images. IPSJ Journal 42, 110-113.

Said, A., Pearlman, W.A., 1996. A new, fast, and efficient image codec based on set partitioning in hierarchical trees. IEEE Trans. Circuits and Systems for Video Technology 6, 243-250.

Shapiro, J.M., 1993. Embedded image coding using zerotrees of wavelet coefficients. IEEE Trans. Signal Process. 41, 3445-3462.

Srinivasan, P., Jamieson, L.H., 1998. High-quality audio compression using an adaptive wavelet packet decomposition and psychoacoustic modeling. IEEE Trans. Signal Process. 46, 1085-1093.

Wang, R.Z., Lin, C.F., Lin, J.C., 2001. Image hiding by optimal LSB substitution and genetic algorithm. Pattern Recognition 34, 671-683.

Captions

Figure 1. Noise-like patch (a) and informative patch (b): (a) complexity 68, (b) complexity 29.

Figure 2. A flowchart of embedding secret data by the proposed method.

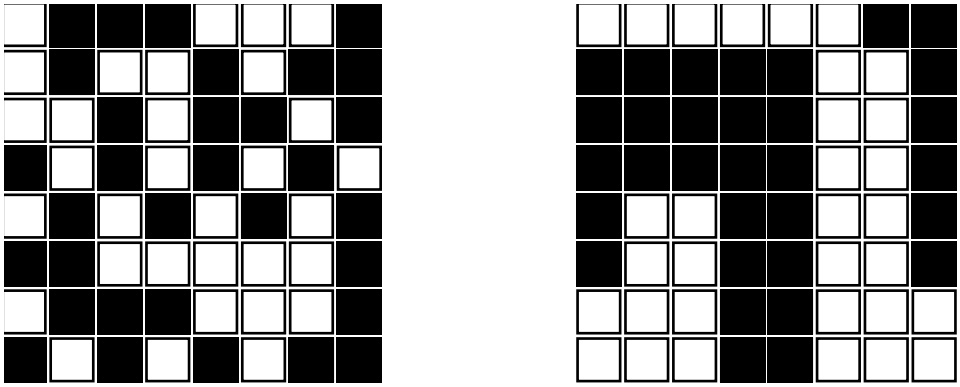
Figure 3. A flowchart of extracting secret data by the proposed method.

Figure 4. Experimental results for "Lena": (a) EZW compressed image (0.59bpp), (b) 8% embedded, (c) 21% embedded and, (d) 33% embedded into (a), (e) EZW compressed image (1.15bpp), (f) 22% embedded, (g) 26% embedded and, (h) 38% embedded into (e).

Figure 5. Experimental results for "Barbara": (a) EZW compressed image (1.17bpp), (b) 9% embedded, (c) 23% embedded and, (d) 37% embedded into (a), (e) EZW compressed image (2.01bpp), (f) 21% embedded (g) 27% embedded and, (h) 43% embedded into (e).

Table 1 Experimental results for "Lena"

Table 2 Experimental results for "Barbara"



(a) (b)

Figure 1: Noise-like patch (a) and informative patch (b): (a) complexity 68, (b) complexity 29.

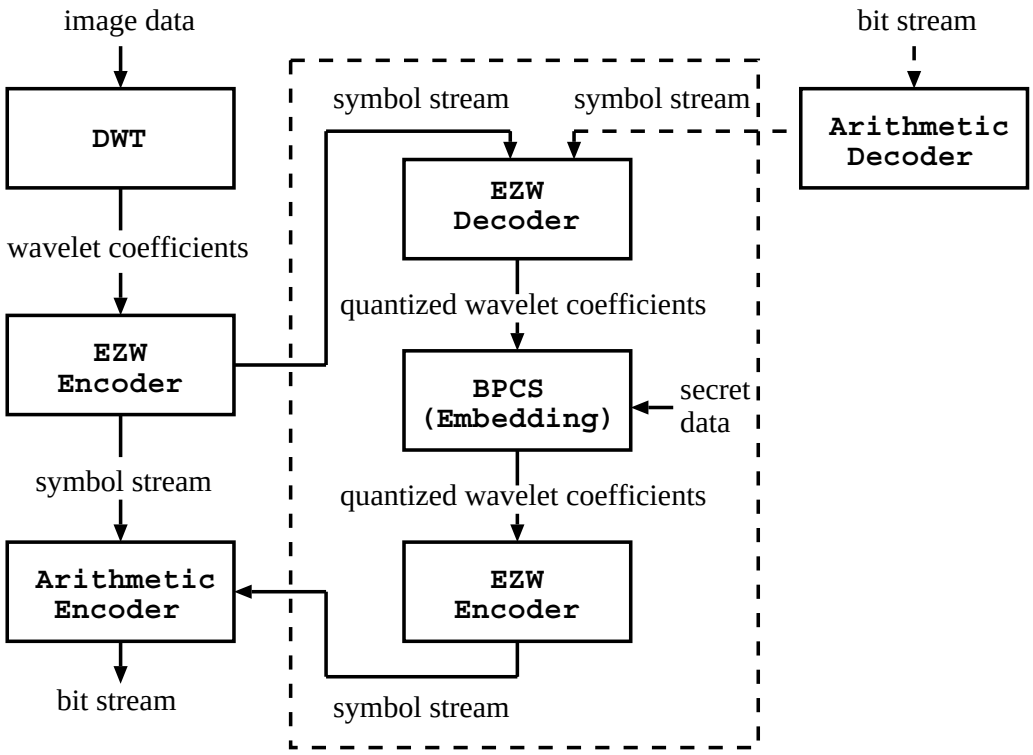


Figure 2: A flowchart of embedding secret data by the proposed method.

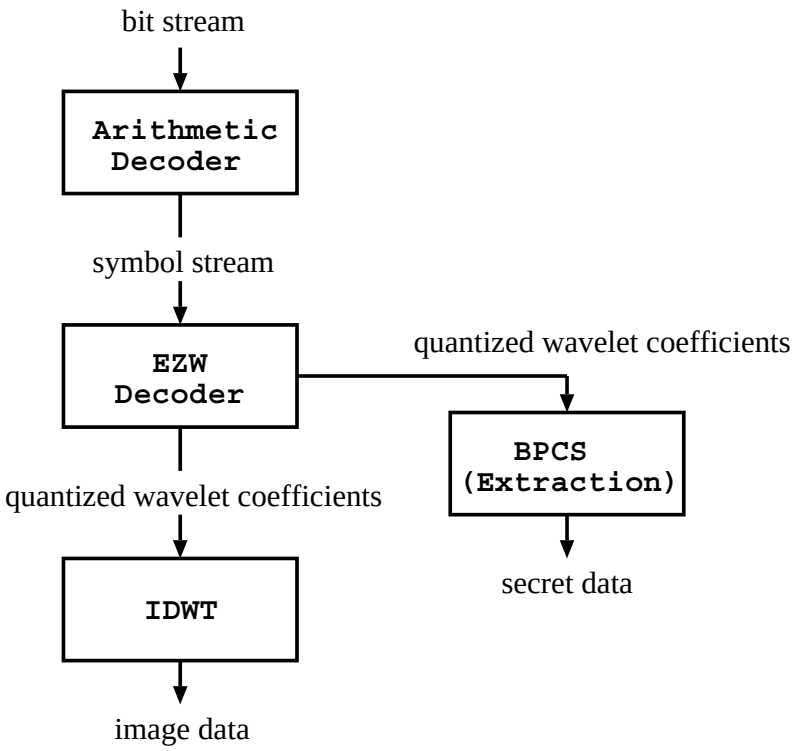
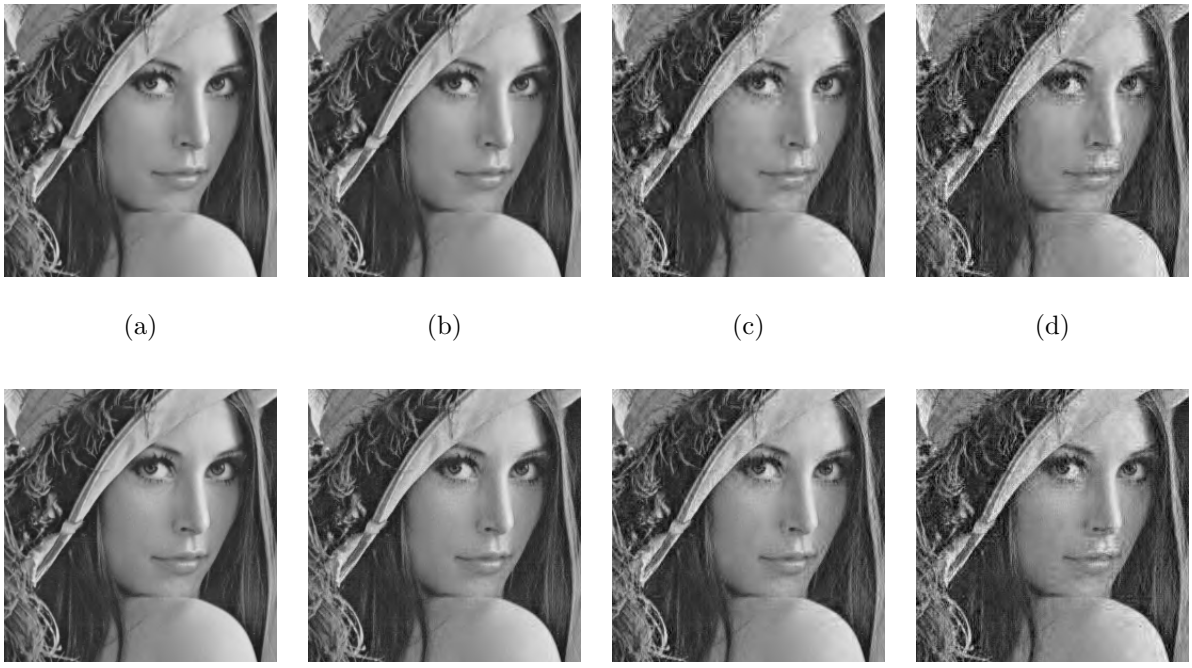


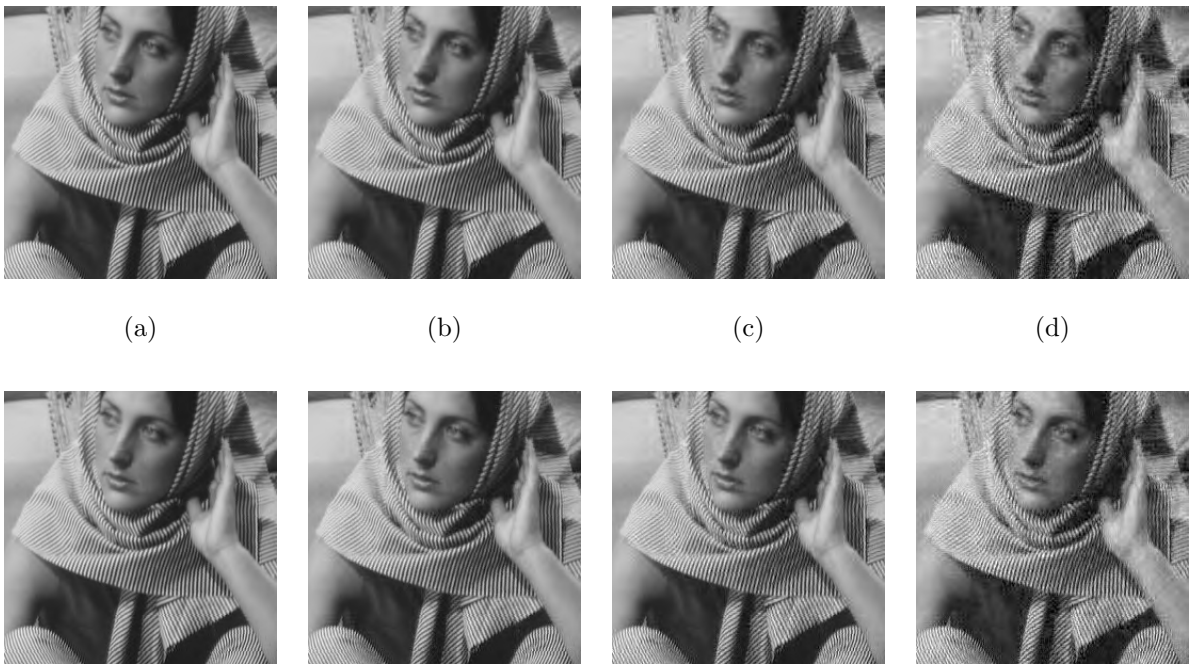
Figure 3: A flowchart of extracting secret data by the proposed method.



(a) (b) (c) (d)

(e) (f) (g) (h)

Figure 4: Experimental results for "Lena": (a) EZW compressed image (0.59bpp), (b) 8% embedded, (c) 21% embedded and, (d) 33% embedded into (a), (e) EZW compressed image (1.15bpp), (f) 22% embedded, (g) 26% embedded and, (h) 38% embedded into (e).



(a) (b) (c) (d)

(e) (f) (g) (h)

Figure 5: Experimental results for "Barbara": (a) EZW compressed image (1.17bpp), (b) 9% embedded, (c) 23% embedded and, (d) 37% embedded into (a), (e) EZW compressed image (2.01bpp), (f) 21% embedded (g) 27% embedded and, (h) 43% embedded into (e).

Table 1 Experimental results for "Lena"

image	# planes	# planes used for embedding	complexity threshold α_0	embedded data (bytes)	compressed file (bytes)	PSNR (dB)
(a)	8	-	-	-	19333	35.9
(b)	8	1	6	1824	22559	35.5
(c)	8	3	6	5104	24219	31.6
(d)	8	3	2	9934	30482	29.1
(e)	9	-	-	-	37630	39.1
(f)	9	2	6	10266	47244	35.6
(g)	9	4	6	12442	47791	33.0
(h)	9	4	2	22908	60032	30.0

Table 2 Experimental results for "Barbara"

image	# planes	# planes used for embedding	complexity threshold α_0	embedded data (bytes)	compressed file (bytes)	PSNR (dB)
(a)	8	-	-	-	38188	34.6
(b)	8	1	8	3768	43191	34.0
(c)	8	3	8	10366	45246	28.9
(d)	8	3	2	21086	56306	26.0
(e)	9	-	-	-	66021	39.2
(f)	9	2	8	16028	76887	34.7
(g)	9	4	8	20546	77476	30.8
(h)	9	4	2	40132	94366	27.1