Sharing Large Secret Messages Using Two AMBTC Shadows

Ching-Chun Chang[1, *] and Ying-Hsuan Huang[2]

[1] Department of Information Management, National Central University
Taoyuan 32001, Taiwan, R.O.C.
4221788@yahoo.com.tw

[2] Department of Computer Science and Engineering, National Chung Hsing University
Taichung 40227, Taiwan, R.O.C.
phd9807@cs.nchu.edu.tw

Received 18 June 2005; Revised 19 February 2006 ; Accepted 20 February 2006

Abstract. To date, several reversible secret sharing methods have been proposed based on absolute moment block truncation coding (AMBTC). These methods can losslessly recover the AMBTC compression codes after revealing the secret data from shadows. However, in these reported methods, one set of AMBTC compression codes, consisting of two quantification values and a bit-plane, is used to share only one secret message. In other words, the embedding capacity of the reported methods can be further improved. In order to share more secret data, we proposed a substitution and mapping method to embed more secret data into a pair of quantification values. In addition, all of the bits in the bit-plane can be used to share secret data. Experimental results showed that the embedding capacity of the proposed method is higher than that of the AMBTC-based secret sharing methods.

Keywords: reversible secret sharing, absolute moment block truncation coding, embedding capacity, bit-plane

1 Introduction

On the Internet, there are three kinds of methods used to protect the transmitted data, i.e., cryptography [1-3], data hiding [4-8], and secret sharing [9-15]. Cryptography encrypts the secret data into the meaningless ciphertext. Although a third party cannot efficiently obtain the secret data, he or she can destroy the ciphertext. In order to solve this problem, the data hiding method was proposed for embedding secret data into a cover medium to obtain the stego medium. Since the stego medium is very similar to the cover medium, the third party cannot efficiently detect the presence of secret messages on the stego medium. However, in the data hiding method, only one user can possess the secret data. Therefore, the secret sharing method was proposed that can decompose the secret data and share them with many participants. Each participant cannot determine all of the secret data from the data that he or she holds. However, by using the data held by all of the participants, the original secret data can be recovered completely, which indicates the high practicality of the secret sharing method.

In 1979, Naor and Shamir [9] proposed a (t, n)-threshold secret sharing method that used a linear polynomial function to share the secret image into n shadows. After collecting the t shadows, the secret image can be obtained by the function of Lagrange basis polynomials. However, there is a significant computational cost associated with the function.

In order to reduce the computational cost, Naor and Shamir [10] proposed a visual cryptography to share the secret image. In the sharing phase, one secret image was encoded as n disorderly and meaningless shadows. Different from the polynomial-based secret sharing method, the secret image can be obtained directly by stacking the t shadows without any computation. However, the secret image cannot be obtained when the number of shadows was less than t. In order to solve the problem, Chao and Lin [11] proposed a progressive secret sharing method. As the number of shadows increases, the visual quality of the secret image was enhanced. However, in the phase of generating the n shadows, the secret image was enlarged, which decreases the visual quality of the secret image.

In 2010, Lin and Chan [12] skillfully shared secret messages by hiding them in meaningful images, thereby protecting the secret data by thwarting the third party's effort to obtain the data. In other words, only the intended participants know that there are some secret messages in the meaningful images. In addition, the original images can be losslessly recovered after extracting the secret data. However, embedding the secret data into meaningful images invokes serious image distortion. In order to solve this problem, Chang *et al*. combined Lin and Chan's scheme and the optimal pixel adjustment process (OPAP) method [13]. In the OPAP method, each pixel was increased or decreased by 2^{n+1} to reduce the distortion of the image, where n denotes the number of secret bits. However, in this kind of method, using meaningful images as the cover media needs more storage spaces.

In order to reduce the storage spaces, Ou and Sun used absolute moment block truncation coding (AMBTC) compression codes as the cover medium, the size of which was smaller than that of the meaningful image. Each set of AMBTC compression codes consists of two quantification values Q_i and a bit-plane BP. The trio $\{Q_1, Q_2, BP\}$ was duplicated to share one secret message. If the secret message s was equal to 0, the two AMBTC compression trios do not need any modification. In other words, the compression codes remain unchanged. Otherwise, if $s = 1$, one of the two trios was flipped by the NOT operator. An example was used to explain the NOT operator. Assume that one bit b in the AMBTC compression codes was 0. After the NOT operator was used, b was inverted as 1. The embedding capacity of the flipping method is equal to the number of sets of AMBTC compression codes.

In order to share more secret messages into the AMBTC compression codes, Chang and Sun proposed a polynomial-based secret sharing scheme [15]. Fig. 1 presents the flowchart of the method. In their method, a pair of quantification values can be used to share one secret message. In addition, the bit-plane was duplicated according to the number of participants, which was denoted by n. After that, the n bits located at the same position of the n bit-planes can be used to share one secret bit. The sharing procedure was to calculate the exclusive-OR (XOR) result of the n bits. If the XOR result was equal to the secret message, the n bits do not need any modification. Otherwise, one of the n bits was flipped, such that the XOR result was equal to the secret message.

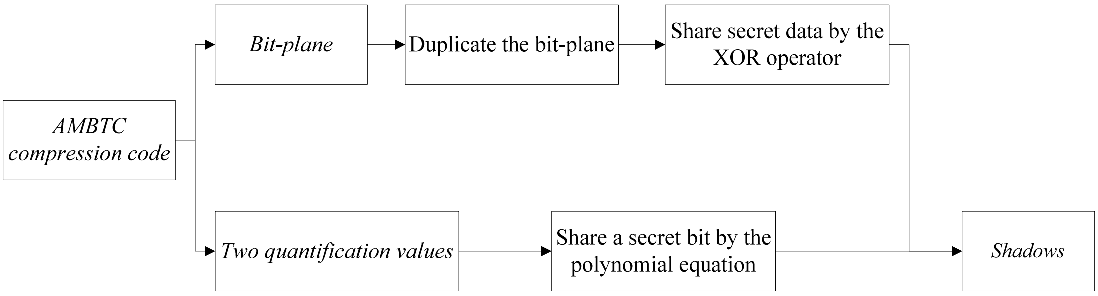

AMBTC compression code
Bit-plane
Duplicate the bit-plane
Share secret data by the XOR operator
Two quantification values
Share a secret bit by the polynomial equation
Shadows

Fig. 1. Flowchart of Chang and Sun's method

In order to share more secret messages into the quantification values, we proposed a substitution and mapping method. Before the sharing phase, the permutations and combinations of the two pairs of quantification values were calculated. According to the secret message and the permutations and combinations, the order between the two pairs of quantification values is rearranged to embed at least two secret bits. The method of embedding the two secret bits has higher embedding capacity than Chang and Sun's method [15]. In addition, the bits in one of the two bit-planes are substituted by the secret bits. The bits in the other bit-plane remain unchanged, and they will be used as the reference for recovering the original AMBTC compression codes.

This paper is organized into five sections. Section 2 introduces two AMBTC-based secret sharing methods. Section 3 describes the proposed method. Section 4 compares the related methods with the proposed method. Our conclusions are given in Section 5.

2 Related Work

2.1 Chang and Sun's method

In Chang and Sun's method [15], one bit-plane with the size of $l \times w$ was duplicated as n bit-planes, where n denotes the number of participants. The n bits b_i ($i = 1, 2, ..., n$) located at the same position of the n bit-planes were used to share one secret bit s by the equation

$$b'_j = \begin{cases} b_j, & \text{if } b_1 \oplus b_2 \oplus ... \oplus b_n = s, \\ |b_j - 1|, & \text{otherwise,} \end{cases} \tag{1}$$

where b_j denotes the bit of the j^{th} bit-plane, and j was determined by a random seed. After the modification procedure, the XOR result was equal to the secret bit. The above procedure shows that the $l \times w$ secret bits can be embedded into the bit-planes.

Both the high mean value HV and the low mean value LV in the AMBTC compression codes were used to share one secret bit by the polynomial

$$f(i) = \begin{cases} HV + LV \times r + a_1 \times r^2 + \ldots + a_{i-2} \times r^i, \text{ if } s = 1, \\ LV + HV \times r + a_1 \times r^2 + \ldots + a_{i-2} \times r^i, \text{ otherwise,} \end{cases} \quad (2)$$

where r is a random number. The polynomial can effectively share one secret bit into $\{HV, LV\}$. Consequently, in Chang and Sun's method, one set of the AMBTC compression codes $\{HV, LV, BP\}$ can be duplicated and used to share $l \times w + 1$ secret bits.

2.2 Ou and Sun's method

Ou and Sun proposed an AMBTC-based data hiding method in 2014 [8]. Fig. 2 shows the flowchart of the method. The traditional AMBTC compression method was used first to encode the compressed block as a bit-plane BP and two quantification values, i.e., the high mean value HV and low mean value LV. Afterwards, the difference d between HV and LV was calculated to determine how many secret bits can be embedded into the compression codes $\{HV, LV, BP\}$. If the difference d was higher than the pre-determined threshold T, one secret bit s was embedded by the following rules.

Rule 1: If $s = 0$, the compression codes remain unchanged, i.e., $\{HV, LV, BP\}$.

Rule 2: If $s = 1$, the compression codes was adjusted as $\{LV, HV, \text{inversed } BP\}$, where the inversed BP was delivered by the NOT operator. Assume that one bit b in BP was 0. By using the NOT operator, b was inverted as 1.

Conversely, if $d \leq T$, all of the bits in BP were directly replaced by secret bits. The replacement does not causes acute image distortion because $HV \approx LV$. According to the modified bit-plane, HV and LV were recalculated by the AMBTC compression method to maintain good visual quality of the decompressed image.

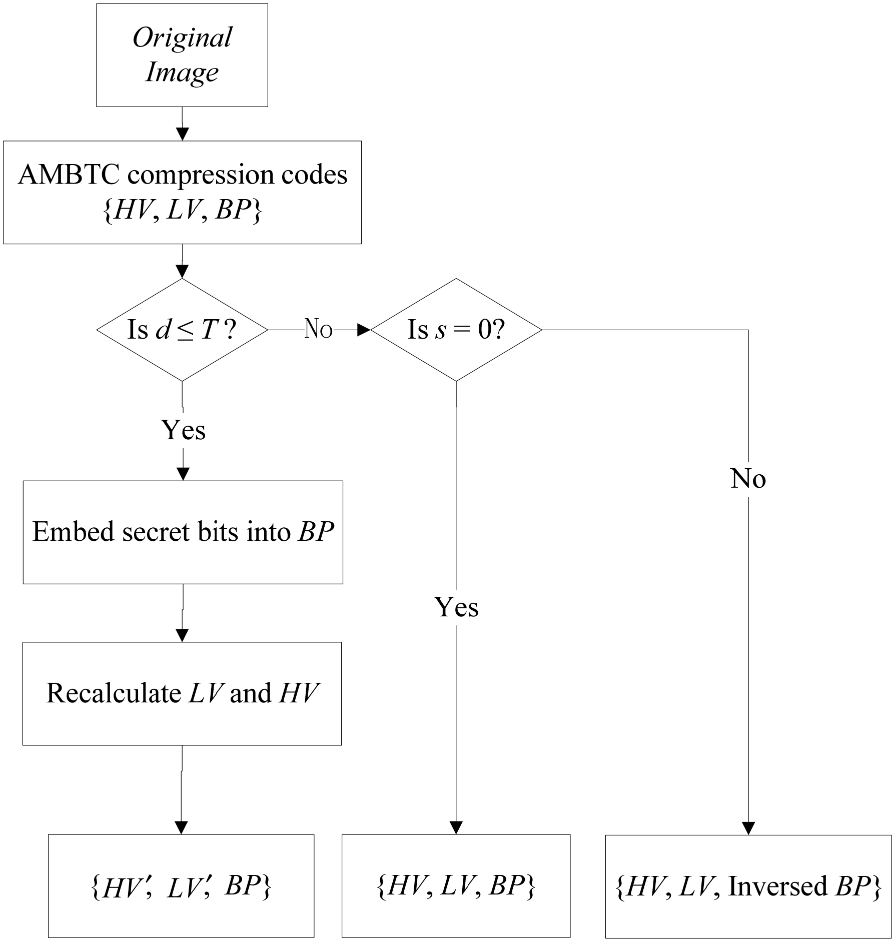


Fig. 2. Flowchart of Ou and Sun's method

3 Proposed Method

The proposed method includes three algorithms, i.e., the AMBTC, secret sharing, and revealing and recovery algorithms. In the proposed method, an original image is divided into $\frac{L \times W}{l \times w}$ blocks, where L and W denote the length and width of the original image, respectively. The notations l and w represent the length and width of the divided block, separately. Each block is further encoded by the AMBTC algorithm to obtain the compression codes $\{HV, LV, BP\}$, where BP consists of $l \times w$ bits. When l and w are enlarged, the compression efficiency of the proposed method is enhanced. However, the visual quality of the decompressed image decreases.

The compression codes $\{HV, LV, BP\}$ are duplicated and used to share massive secret messages by the substitution and mapping method. In the revealing and recovery phase, all of the secret data can be extracted correctly and the original compression codes can be recovered. Fig. 3 shows the flowchart of the proposed method.

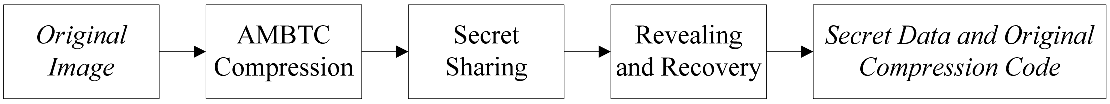


Fig. 3. Flowchart of the proposed method

3.1 AMBTC Algorithm

In Section 3.1, the original image is divided into $\frac{L \times W}{l \times w}$ blocks, each of which is further compressed as $\{HV, LV, BP\}$ by the following procedure. First of all, the average value AV of the $l \times w$ original pixels in the block is used to compress the original pixel P_i as one compression bit b_i, i.e.,

$$b_i = \begin{cases} 0, \text{ if } P_i \le AV, \\ 1, \text{ otherwise,} \end{cases} \text{ where } AV = \left\lfloor \frac{1}{l \times w} \sum_{i=1}^{l \times w} P_i \right\rfloor . \tag{3}$$

After the above procedure, the $l \times w$ compression bits form BP. The number of compression bits b_i's that have the value of 1 is counted to obtain the quantity of the larger pixels, i.e., $n = \sum_{i=1}^{l \times w} b_i$. According to n and P_i, the high mean value of the larger pixels is derived by the equation

$$HV = \left\lfloor \frac{1}{n} \sum_{i=1}^{n} P_i \right\rfloor, \text{ where } P_i > AV. \tag{4}$$

Conversely, the low mean value of the smaller pixels is derived by the equation

$$LV = \left\lfloor \frac{1}{l \times w - n} \sum_{i=1}^{l \times w - n} P_i \right\rfloor, \text{ where } P_i \le AV. \tag{5}$$

The above procedure is repeated until all blocks are compressed.

3.2 Sharing Algorithm

In order to share more secret bits, the set of AMBTC compression codes $\{HV, LV, BP\}$ is duplicated, e.g., $\{HV_1, LV_1, BP_1\}$ and $\{HV_2, LV_2, BP_2\}$. Note that, if $HV = LV$, both $\{HV_1, LV_1, BP_1\}$ and $\{HV_2, LV_2, BP_2\}$ cannot be used to embed any secret message. Conversely, if $HV \neq LV$, both $\{HV_1, LV_1, BP_1\}$ and $\{HV_2, LV_2, BP_2\}$ are used to share at least $l \times w + 2$ secret bits. The sharing procedure is as follows.

Both $\{HV_1, LV_1\}$ and $\{HV_2, LV_2\}$ are used to share at least two secret bits by the permutations and combinations. Table 1 lists the permutations and combinations of $\{HV_1, LV_1\}$ and $\{HV_2, LV_2\}$, which includes six combinations. Consequently, the proposed method can embed a six-based secret value by swapping four quantification values $\{HV_1, LV_1, HV_2, LV_2\}$. Assume that the six-based secret value is $(2)_6$. The two pairs of quantification values $\{HV_1, LV_1\}$ and $\{HV_2, LV_2\}$ are swapped as $\{HV_1, HV_2\}$ and $\{LV_1, LV_2\}$.

In order to further embed more secret data, both BP_1 and BP_2 are used to share $l \times w$ secret bits $\{sb_1, sb_2, ..., sb_{l \times w}\}$. The sharing procedure is as follows. Both HV and LV are converted into the $l \times w$ binary bits $\{bv_1, bv_2, ..., bv_{l \times w}\}$. Then, the bit b_i in the bv_ith bit-plane is directly replaced by sb_i. Meanwhile, the bit in the other bit-plane remains unchanged, and it will be used to recover the original compression codes.

Table 1. Permutations and combinations of $\{HV_1, LV_1\}$ and $\{HV_2, LV_2\}$

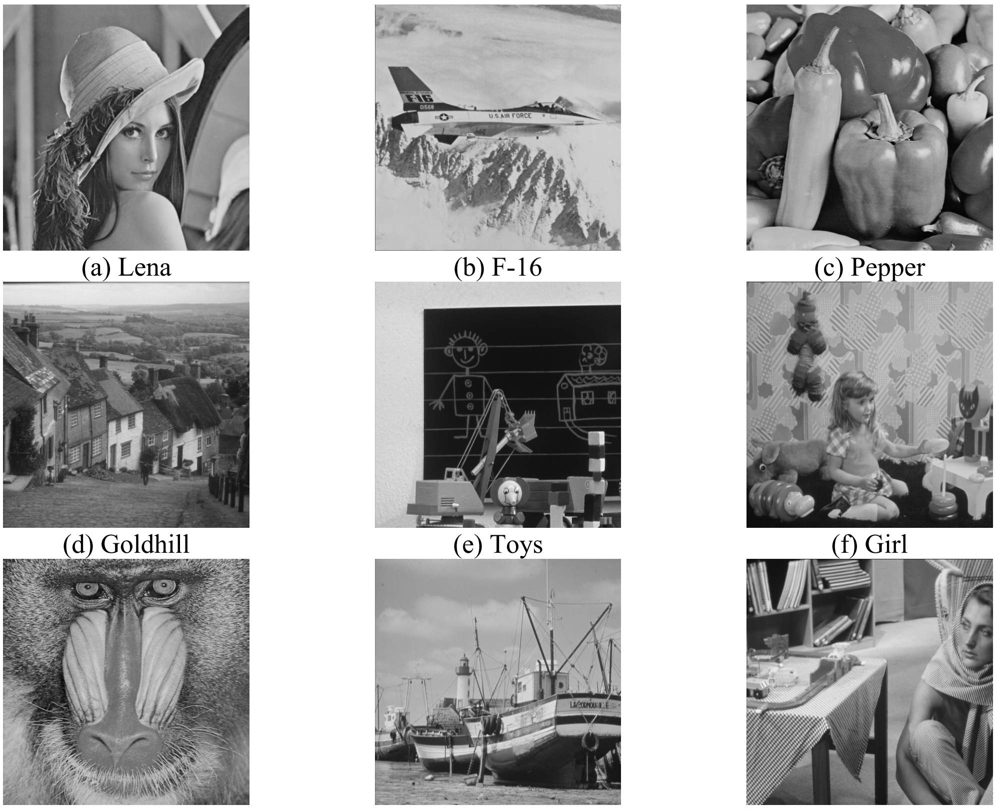
Six-based secret value	Quantification values	
	First pair	Second pair
1	$\{HV_1, LV_1\}$	$\{HV_2, LV_2\}$
2	$\{HV_1, HV_2\}$	$\{LV_1, LV_2\}$
3	$\{LV_1, LV_2\}$	$\{HV_1, HV_2\}$
4	$\{HV_1, LV_1\}$	$\{LV_2, HV_2\}$
5	$\{LV_1, HV_1\}$	$\{HV_2, LV_2\}$
6	$\{LV_1, HV_1\}$	$\{LV_2, HV_2\}$

3.3 Revealing and Recovery Algorithm

In Section 3.3, the embedded secret data can be revealed correctly and the original compression codes $\{HV, LV, BP\}$ can be recovered. First of all, the six-based secret value is extracted by mapping the two pairs of quantification values into the secret value of Table 1. For example, both $\{HV_1, HV_2\}$ and $\{LV_1, LV_2\}$ are mapped into the six-based secret value $(2)_6$. Afterwards, the $l \times w$ secret bits embedded in the two bit-planes are extracted by the following procedure. Both HV_1 and LV_1 are converted into the $l \times w$ binary bits $\{bv_1, bv_2, ..., bv_{l \times w}\}$. The bit in the bv_i^{th} bit-plane is just the embedded secret bit. In addition, the bit in the other bit-plane is just the original compression bit. After the above procedures, all of the embedded bits are extracted correctly and the entire AMBTC compression codes $\{HV, LV, BP\}$ are recovered.

4 Experimental Results

Fig. 4 shows nine grayscale images, which were used as test images. During the experimental phase, the pure embedding capacity was used to compare the proposed method and the AMBTC-based secret sharing methods [7, 8, 15]. Since the AMBTC compression codes of the proposed method were duplicated to share the secret messages, the embedding capacity of the proposed method was divided by 2 to obtain the pure embedding capacity.



(a) Lena (b) F-16 (c) Pepper

(d) Goldhill (e) Toys (f) Girl

(g) Baboon (h) Boat (i) Barbara

Fig. 4. Nine test images

Fig. 5 presents the pure embedding capacity of the proposed method with different sizes of the block. The pure embedding capacity increases as the size of the compressed block decreases. This is because the original image was divided into more smaller-sized blocks. Each of the divided blocks was encoded as $\{HV, LV, BP\}$, which can be used effectively to embed $l \times w + 2$ secret bits. Conversely, as the size of the block increases, the number of AMBTC compression codes decreases, thereby reducing the pure embedding capacity.

Fig. 5 shows that the pure embedding capacity of the image "F-16" is significantly lower than that of the image "Baboon." This is because there is a high probability that HV is equal to LV in the smooth region of the image "F-16." If $HV = LV$, the compression codes $\{HV, LV, BP\}$ cannot be used to share any secret bit. However, for the entire image, the probability of $LV = HV$ is low. Consequently, most compression codes can be used to share secret data.

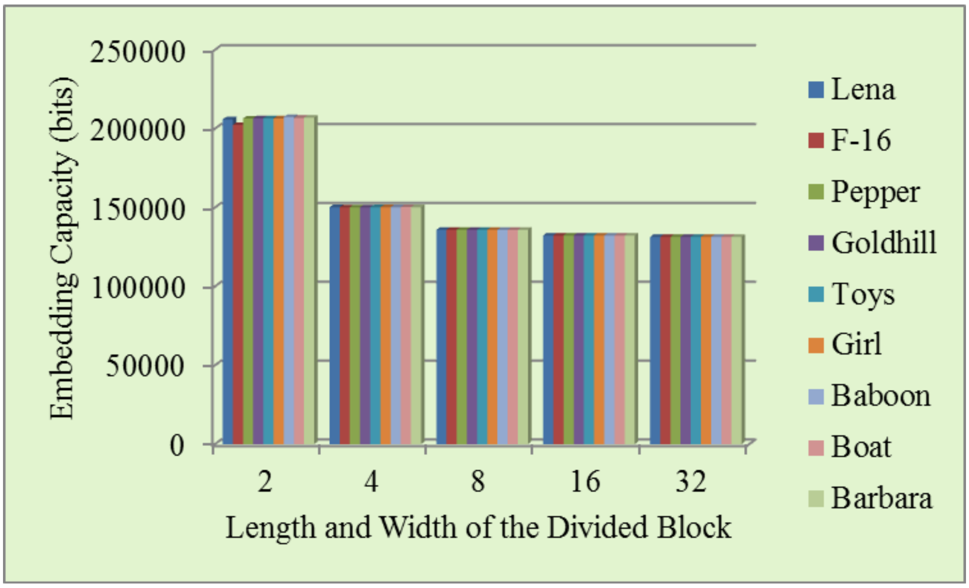


Fig. 5. Embedding capacity of the proposed method with different sizes of the block

Table 2 shows that the pure embedding capacity of the proposed method is significantly higher than that of Ou and Sun's method [8]. This is because their method cannot effectively embed secret data into all bit-planes in order to recover the original compression codes. In other words, only a few bit-planes for which the differences between HV and LV are 0 are used to embed secret data. As a result, the embedding capacity of Ou and Sun's method is lower than that of the proposed method.

On the other hand, the proposed method can share more secret messages than Chang and Sun's method [15]. This is because, in Chang and Sun's method, the pair of quantification values is used to share only one secret message. Different from their method, the proposed method can duplicate the pair of quantification values and share at least two secret bits into them; that is why the proposed method can achieve high embedding capacity.

Table 2. Comparison of the proposed method with those of the related methods [8, 15] in terms of pure embedding capacity (bits)

Methods	Images								
	Lena	F-16	Pepper	Goldhill	Toys	Girl	Baboon	Boat	Barbara
Our	150,195	150,151	150,165	150,089	150,175	150,177	150,175	150,253	150,202
[8]	16,399	16,504	16,384	16,549	16,384	16,384	16,384	16,384	16,384
[15]	139,264	139,264	139,264	139,264	139,264	139,264	139,264	139,264	139,264

Table 3 shows that the pure embedding capacity of the proposed method is higher than that of Lo *et al.*'s method [7]. This is because Lo *et al.* do not embed any secret message into the bit-plane. Conversely, in the proposed method, all of the bits in the bit-plane were used to share secret data. Therefore, the pure embedding capacity of the proposed method is superior to that of Lo *et al.*'s method.

Table 3. Comparison of pure embedding capacity between the proposed method and Lo *et al*.'s method [7]

Methods	Images						
	Lena	F-16	Pepper	Goldhill	Toys	Girl	Boat
Our	4,025	7,073	4,882	2,410	6,189	3,518	5,387
[7]	150,195	150,151	150,165	150,089	150,175	150,177	150,253

5 Conclusions

In this paper, we proposed a substitution and mapping method to share a large amount of secret messages into the AMBTC compression codes. In the proposed method, all of the bits in the bit-plane can be used to share secret data. In addition, a pair of quantification values in the AMBTC compression codes can be duplicated and used to share at least two secret bits. In other words, the AMBTC compression codes are used effectively to share massive secret data. After extracting the secret data, the original AMBTC compression codes can be recovered. Experimental results showed that the pure embedding capacity of the proposed method is significantly higher than that of the other methods. The merits indicate the high practicality of the proposed method.

References

[1] R. Rivest, A. Shamir, L. Adleman, "A method for obtaining digital signatures and public-key cryptosystems," *Communications of the Association for Computing Machinery*, Vol. 21, No. 2, pp. 120-126, 1978.

[2] J. Cui, L. Huang, H. Zhong, C.C. Chang, W. Yang, "An improved AES S-box and its performance analysis," *International Journal of Innovative Computing, Information and Control*, Vol. 7, No. 5, pp. 2291-2302, 2011.

[3] L. Harn, C.C. Chang, H.L. Wu, "An anonymous multi-receiver encryption based on RSA," *International Journal of Network Security*, Vol. 15, No. 4, pp. 307-312, 2013.

[4] C. Qin, C.C. Chang, Y.H. Huang, L. T. Liao, "An inpainting-assisted reversible steganographic scheme using a histogram shifting mechanism," *IEEE Transactions on Circuits and Systems for Video Technology*, Vol. 23, No. 7, pp. 1109-1118, 2013.

[5] T.C. Lu, C.Y. Tseng, J.H. Wu, "Dual imaging-based reversible hiding technique using LSB matching," *Signal Processing*, Vol. 108, pp. 77-89, 2015.

[6] T.C. Lu, C.Y. Tseng, K.M. Deng, "Reversible data hiding using local edge sensing prediction methods and adaptive thresholds," *Signal Processing*, Vol. 104, pp. 152-166, 2014.

[7] C.C. Lo, Y.C. Hu, W.L. Chen, C.M. Wu, "Reversible data hiding scheme for BTC-compressed images based on histogram shifting," *International Journal of Security and Its Applications*, Vol. 8, No. 2, pp. 301-314, 2014.

[8] D. Ou and W. Sun, "High payload image steganography with minimum distortion based on absolute moment block truncation coding," *Multimedia Tools and Applications*, 2014. (DOI: 10.1007/s11042-014-2059-2)

[9] A. Shamir, "How to share a secret," *Communications of the Association for Computing Machinery*, Vol. 22, No. 11, pp. 612-613, 1979.

[10] M. Naor and A. Shamir, "Visual cryptography," *Lecture Notes in Computer Science*, Vol. 950, pp. 1-12, 1995.

[11] K.Y. Chao and J.C. Lin, "User-friendly sharing of images: progressive approach based on modulus operations," *Journal of Electronic Imaging*, Vol. 18, No. 3, 2009.

[12] P.Y. Lin and C.S. Chan, "Invertible secret image sharing with steganography," *Pattern Recognition Letters*, Vol. 31, No. 13, pp. 1887-1893, 2010.

[13] C.C. Chang, Y.H. Huang, T.C. Liu, "Reversible secret sharing with distortion control mechanism," *Innovative Computing Information and Control--Express Letters, Part B: Applications*, Vol. 1, No. 2, pp. 163-167, 2010.

[14] D. Ou and W. Sun, "Reversible AMBTC-based secret sharing scheme with abilities of two decryptions," *Journal of Visual Communication and Image Representation*, Vol. 25, No. 5, pp. 1222-1239, 2014.

[15] C.C. Chang and C.Y. Sun, "Polynomial-based secret sharing scheme based on the absolute moment block truncation coding technique," *Proceedings of the Tenth International Conference on Intelligent Information Hiding and Multimedia Signal Processing*, pp. 485-488, 2014.