Multimed Tools Appl (2018) 77:18067–18083
DOI 10.1007/s11042-017-4957-6

Reversible data hiding in encrypted AMBTC images

Zhaoxia Yin[1,2] · **Xuejing Niu**[1] · **Xinpeng Zhang**[2] · **Jin Tang**[1] · **Bin Luo**[1]

Received: 15 March 2017 / Revised: 21 May 2017 / Accepted: 16 June 2017 /
Published online: 16 August 2017
© Springer Science+Business Media, LLC 2017

Abstract Signal processing in the encrypted domain has attracted a lot of attention due to the requirement for content security and privacy protection. Reversible data hiding in encrypted images (RDH-EI) is also a hot topic. However, the majority of the published techniques are designed for uncompressed images rather than JPEG-, VQ- and BTC-compressed images. In this paper, for the first time, a RDH-EI method for AMBTC images is proposed. In the proposed method, the higher mean and lower mean of a triple in an AMBTC-compressed image are encrypted by using stream cipher at first. Then, additional data can be embedded into the redundant space by using prediction error histogram modification technique. Experimental results and analysis demonstrate that, with the marked cipher-image, legal receivers are able to extract embedded data exactly by using a data hiding key, decrypt it to recover an image very similar to the original one by using an image encryption key, or extract additional data and recover the original image error free with both keys. The proposed method is applicable to real-time transmission due to the simple implementation of the algorithm and low computational complexity.

Keywords Signal processing in encrypted domain (SPED) · Privacy protection · Absolute moment block truncation coding (AMBTC) · Reversible data hiding in encrypted images (RDH-EI)

1 Introduction

As one of the most significant information security research topic, data hiding is a technique that embeds additional data into a host signal while leaving little distortion,

✉ Bin Luo
luobin_ahu@163.com

[1] Key Laboratory of Intelligent Computing & Signal Processing, Ministry of Education, Anhui University, Hefei 230601, People's Republic of China

[2] School of Communication and Information Engineering, Shanghai University, Shanghai 200072, People's Republic of China

which can be classified into non-reversible [6, 28] and reversible data hiding (RDH) [18, 19]. According to different applications, it is often called watermarking [24] for copyright protection [10, 22, 33] or steganography [21] for convert communication. Steganography aims to embed additional data into digital media secretly by slightly modifying the cover data, while steganalysis [23, 25] attempts to reveal the presence of the embedded data. For reversible data hiding, the original cover signal can be recovered error-free and it is often achieved based on difference expansion (DE) [20] or histogram shifting (HS) [11]. For plaintext images, all these methods have good embedding efficiency. However, the original cover images have to be accessible to or seen by the data hider during the embedding process.

The needs of content security and privacy protection contribute to the emergence of signal processing in the encrypted domain (SPED) [2–5]. Reversible data hiding in encrypted images (RDH-EI), as one of the most commonly researched SPED topics, enables additional data to be embedded into a cipher-image without revealing the original image, and also allows the original image to be recovered error-free at the receiver side [13, 29].

The first RDH-EI method was proposed in [13]. In this method, the original image is encrypted with Advanced Encryption Standard (AES), and one bit can be embedded into a block consisting of n pixels. Therefore, the embedding rate of this method achieves $1/n$ bpp. On the receiver side, after decryption of the marked cipher-image, the analysis of the local standard deviation is indispensable for data extraction and image recovery. This is a good idea. However, the quality of the decrypted image is rather poor and far from human vision requirements if the receiver decrypts the marked encrypted image directly.

In 2011, an entirely new idea and method was proposed [29], in which the image owner can encrypt the original image by using a stream cipher and one bit additional data is embedded into cipher-image by flipping the three least significant bits (LSB) of pixels in each non-overlapping block. On the receiver side, the marked cipher-image is decrypted to obtain an image which is approximate to the original image. To form a new block, three LSBs of pixels in each block are flipped by the receiver and a function is used to estimate each block's image-texture. The modified block is assumed to be rougher than the original block due to the spatial correlation of natural images. Therefore, the receiver can extract embedded bits and recover the original image jointly.

This well-known method has been the focus of great attention [7, 17, 32]. Hong et al. improved the performance by utilizing a side-match method to enhance the embedding rate while leaving lower error rates during the image recovery phase [7]. Furthermore, during data embedding in Qin and Zhang's method [17], rather than flip the LSBs of half of the pixels, they flip the LSBs of fewer pixels, leading to significant visual improvement of the approximate image. To estimate the image-texture of each block, they utilize a new adaptive judging function, which is based on the distribution characteristic of local image contents, for the procedures of data extraction and image recovery. Consequently, to some extent, the errors of extracted bits and the recovered image decrease. It is obvious that the embedding rate of these methods largely depends on the size of each block. In other words, errors potentially appear during data extraction and image recovery with inappropriate block sizes. Recently, a RDH-EI method, which is based on a public key cryptosystem and

homomorphic encryption, has been proposed [32]. In this method, cipher images are encrypted by public key cryptosystems with probabilistic and homomorphic properties. It is a lossless, reversible, and combined data hiding method for cipher images. In this lossless method, new values substitute for the cipher pixels so that additional bits can be embedded into several cipher pixels' LSB-planes by using multi-layer wet paper coding. Afterwards, the embedded data are able to be extracted from the encrypted image directly, and the decryption of original image will not be affected by the data embedding operation. In this reversible method, a pretreatment, which is used to shrink the histogram of image before image encryption phase, is adopted. Therefore, no pixel oversaturation of plaintext images would happen when the encrypted images are modified to embed data. From the directly decrypted image, embedded data can still be extracted and the original content recovered when slight distortion is induced. Since the lossless and reversible methods are compatible, these two methods can be performed simultaneously in data embedding of encrypted image. Therefore, on the receiver side, a part of the embedded data can be extracted before decryption, another part of the embedded data can be extracted, and the original image can be recovered after decryption.

Up to now, the existing RDH-ED methods can be classified into two categories: vacating room before encryption [1, 31] and vacating room after encryption [8, 12, 15, 26, 27, 30]. However, all of the methods introduced above are focus on uncompressed images. To the best of our knowledge, there are few RDH-EI methods designed for compressed images such as JPEG (Joint Photographic Experts Group), BTC (Block Truncation Coding) and VQ (vector quantization). For JPEG images, Qian et al. proposed several methods recently [14, 16], in which additional data are embedded into the encrypted JPEG bit-stream, and the original bit-stream can be recovered by means of analyzing blocking artifacts induced by data hiding. In this paper, a RDH-EI method designed for AMBTC compressed images is proposed. AMBTC was proposed by Lema and Mitchell [9]. Compared with VQ and JPEG, AMBTC is more specifically suited to real-time application and less powerful processing application due to its lower computational complexity. It has been the subject of development in the real-time image transmission field. WSN (Wireless Sensor Networks), as low-power processor, needs not change the battery for at least 5 years or merely uses the solar energy. But WSN only has low bandwidth, low-end processors, and small amount of storage. That means AMBTC is required. The technique of AMBTC is described in detail in Section 2.

The proposed AMBTC RDH-EI method consists of three phases: image encryption, data embedding, and data extraction and image recovery. In the data embedding phase, the higher mean and lower mean of a triple in AMBTC-compressed image are encrypted by a stream cipher with the same random bits. Therefore, the block correlation of a natural image is preserved and the redundant space can be exploited by using the histogram of the prediction error in the data embedding phase. Experimental results and analysis demonstrate that, with the marked cipher-image, legal receivers are able to extract the embedded data exactly by using a data hiding key, decrypt it to get an image very similar to the original one by using an image encryption key, or extract additional data and recover the original image error free with both keys. In addition, the proposed method is applicable to real-time transmission due to the simple algorithm and low computational complexity.

2 AMBTC compression technique

To compress a gray-scale image I sized $H \times W$, I is subdivided into non-overlapping $m \times n$ blocks. In total, $N = \lfloor H/m \rfloor \times \lfloor W/n \rfloor$ blocks can be obtained. We denote $p_{i,j}$ as the j-th pixel of block P_i. Therefore, the i-th block is represented as $P_i = \{p_{i,1}, p_{i,2}, \cdots, p_{i,m\times n}\}$. The mean of P_i can be calculated as

$$\overline{P}_i = \frac{1}{m \times n} \sum_{j=1}^{m\times n} p_{i,j} \tag{1}$$

The bit plane $b_i = \{b_{i,1}, b_{i,2}, \cdots, b_{i,m\times n}\}$ is generated to record the comparative result between $p_{i,j}$ and $\overline{P}_i$

$$b_{i,j} = \begin{cases} 0, p_{i,j} < \overline{P}_i \\ 1, p_{i,j} \geq \overline{P}_i \end{cases} \tag{2}$$

For the i-th block, the lower mean l_i is the average of pixels whose values are smaller than $\overline{P}_i$. Similarly, the higher mean h_i is the average of pixels whose values are not smaller than $\overline{P}_i$. Therefore, the i-th block in I can be compressed into triple (l_i, h_i, $b_i = \{b_{i,1}, b_{i,2}, \cdots, b_{i,m\times n}\}$), and the AMBTC-compressed image $\tilde{I}$ can be represented as $\{l_i, h_i, b_i\}_{i=1}^{N}$.

To decode AMBTC-compressed image $\tilde{I}$, each triple in $\{l_i, h_i, b_i\}_{i=1}^{N}$ is visited. Suppose (l_i, h_i, $b_i = \{b_{i,1}, b_{i,2}, \cdots, b_{i,m\times n}\}$) is decoded as $\tilde{P}_i = \{\tilde{p}_{i,1}, \tilde{p}_{i,2}, \cdots, \tilde{p}_{i,m\times n}\}$. If $b_{i,j} = 0$, then $\tilde{p}_{i,j} = l_i$. Otherwise, if $b_{i,j} = 1$, then $\tilde{p}_{i,j} = h_i$. After all block are processed in the same way, the AMBTC-compressed image can be obtained.

3 Proposed method

The detail of the proposed method is presented in this section. The proposed method is divided into three phases: image encryption, data embedding, and data extraction and image recovery. The diagram of the proposed method is shown in Fig. 1. The image owner encrypts an AMBTC-compressed image with an image encryption key. Then, using a data-hiding key, the data owner can embed additional data into the encrypted AMBTC image in the absence of the original image content. On the receiver side, with the marked cipher AMBTC image which contains additional data, a legal receiver is able to extract the embedded data by using the data-hiding key, or decrypt it directly by using the image encryption key, or extract data and recover the original AMBTC image error-free by using both keys.

3.1 Image encryption

Stream cipher is a typical encryption method and is commonly used in image encryption [8, 30]. Supposing the AMBTC image $\tilde{I}$ consists of N triples. For the i-th block, the triple is (l_i, h_i, $b_i = \{b_{i,1}, b_{i,2}, \cdots, b_{i,m\times n}\}$). It is obvious that l_i and h_i fall into [0, 255]. Denote the bits of l_i as $l_{i,0}$, $l_{i,1}$, $\cdots$, $l_{i,7}$. Thus

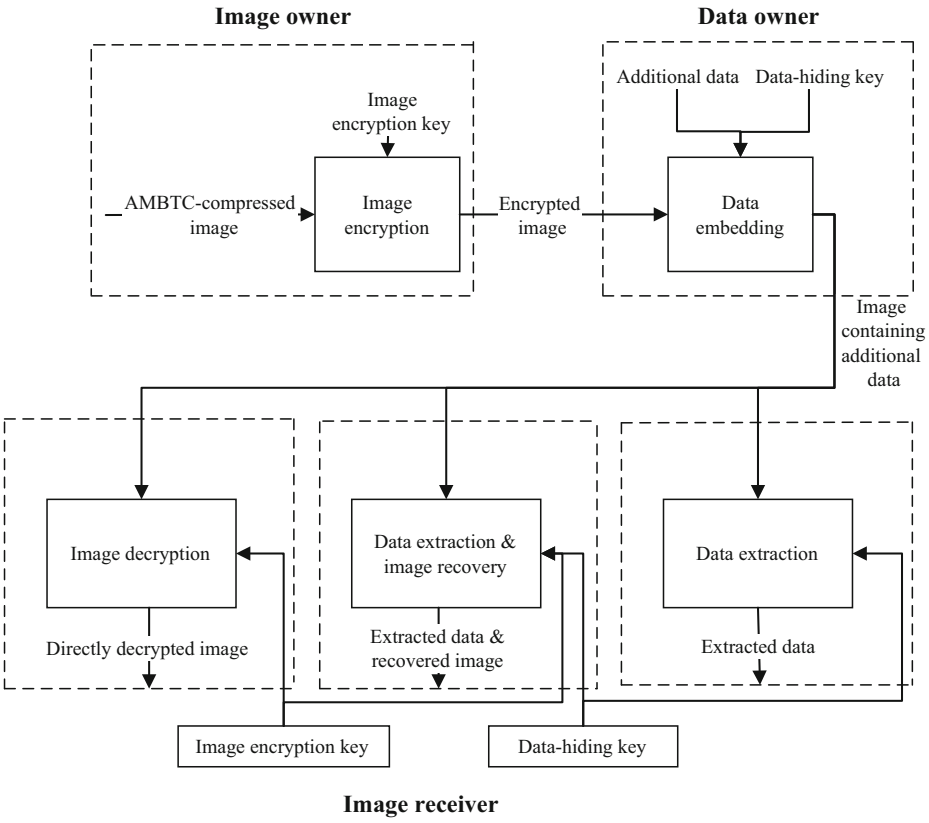

Fig. 1 Diagram of the proposed method

$$l_{i,j} = \left\lfloor \frac{l_i}{2^j} \right\rfloor \bmod 2, j = 0, 1, \cdots 7 \tag{3}$$

Similarly

$$h_{i,j} = \left\lfloor \frac{h_i}{2^j} \right\rfloor \bmod 2, j = 0, 1, \cdots 7 \tag{4}$$

Then encrypt $l_{i,j}$ and $h_{i,j}$ by using exclusive or operation. For h_i

$$H_{i,j} = h_{i,j} \oplus r_{i,j}, j = 0, 1, \cdots, 7 \tag{5}$$

Similarly

$$L_{i,j} = l_{i,j} \oplus r_{i,j}, j = 0, 1, \cdots, 7 \tag{6}$$

Where $r_{i,j}$ is pseudo-random binary number determined by image encryption key. Note that the random binary numbers used to encrypt $l_{i,j}$ and $h_{i,j}$ are the same. The correlation between $l_{i,j}$ and $h_{i,j}$ still exists for the existence of tradeoff between encryption and data embedding. A fully encrypted image has theoretically reached

maximum information entropy, so embedding data in it is impractical. The gray values of the encrypted higher mean and lower mean are

$$H_i = \sum_{j=0}^{7} H_{i,j} \cdot 2^j \tag{7}$$

And

$$L_i = \sum_{j=0}^{7} L_{i,j} \cdot 2^j \tag{8}$$

Then to encrypt $b_i = \{b_{i,1}, b_{i,2}, \cdots, b_{i,m\times n}\}$, pseudo-random binary number $s_{i,k}$, which is also generated by the image encryption key, is adopted. Here

$$B_{i,k} = b_{i,k} \oplus s_{i,k}, k = 0, 1, \cdots, m \times n \tag{9}$$

The encrypted b_i is $B_i = \{B_{i,1}, B_{i,2}, \cdots, B_{i,m\times n}\}$. Accordingly, the i-th encrypted triple is represented as $(H_i, L_i, B_i = \{B_{i,1}, B_{i,2}, \cdots, B_{i,m\times n}\})$. Both H_i, L_i together with B_i are encrypted with random binary numbers, therefore, for an attacker without an encryption key, it's impractical to reveal the content of the plain image. The AMBTC image $\tilde{I}$, consisting of N triples, is encrypted triple by triple, hence, time complexity of image encryption process is $O(N)$.

3.2 Data embedding

When the AMBTC image is encrypted, the data hider has to embed additional data into the cipher image. For the triple (H_i, L_i, B_i), we utilize B_i and modify L_i to create the embedding space which can embed a considerable number of bits. We define the prediction error as

$$PE_i = H_i - L_i, PE_i \in [-255, 255] \tag{10}$$

By calculating all triples, a histogram of PE can be obtained. The values of H_i and L_i are close in most cases because of the correlation to the original image. Most of PE are close to 0, hence the peak points of the histogram are exploited to embed data. Two peak points in the histogram of PE can be utilized. As shown in Figs. 2 and 3, bin −2 and bin 3 are expanded to embed data, bins larger than 3 or smaller than −2 are shifted, and bins between −2 and 3 remain unchanged. Supposing two peak points are P_1 and P_2 respectively, and corresponding zero points are Z_1 and Z_2. Here $Z_1 < P_1 < P_2 < Z_2$. If H_i is equal to L_i, we replace the bit plane with binary additional data to be embedded directly, and set $flag_i = 1$ at the same time, otherwise, set $flag_i = 0$. Then modify L_i according to the following equation:

$$L_i' = \begin{cases} L_i + 1, Z_1 < PE_i < P_1 \\ L_i + d, PE_i = P_1 and flag_i \neq 1 \\ L_i - d, PE_i = P_2 and flag_i \neq 1 \\ L_i - 1, P_2 < PE_i < Z_2 \end{cases} \tag{11}$$

Here $d \in \{0, 1\}$ represents one binary additional data. $flag_i$ is merely used for recording whether the bit plane in the i-th triple is replaced by additional data in the embedding phase.

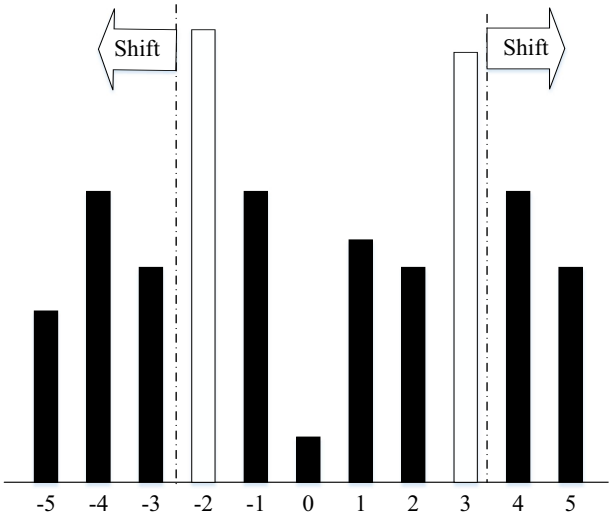

Fig. 2 The histogram of *PE*

Note that L_i may be modified in the embedding phase. Overflow and underflow problems are likely to happen. To solve this problem, a location map is adopted here. If L_i is 255 or 0, the corresponding location map point will be set as 1 and L_i will be modified to be 254 or 1. If L_i is 254 or 1, the corresponding location map point will be set as 0 and L_i remains unchanged. The embedding technique should be carried out after overflow and underflow processing so that location map can be embedded into host image and needs not to be stored by extra devices.

Returning to the process of data embedding, after location map is obtained and L_i which is equal to 255 or 0, is revised as 254 or 1, 4 cases must be considered in data embedding of the i-th triple:

Case 1: if $PE_i = 1$, $L_i = 254$ and corresponding location map is 1, then the bit plane should be replaced by binary additional data to be embedded and $flag_i = 1$;

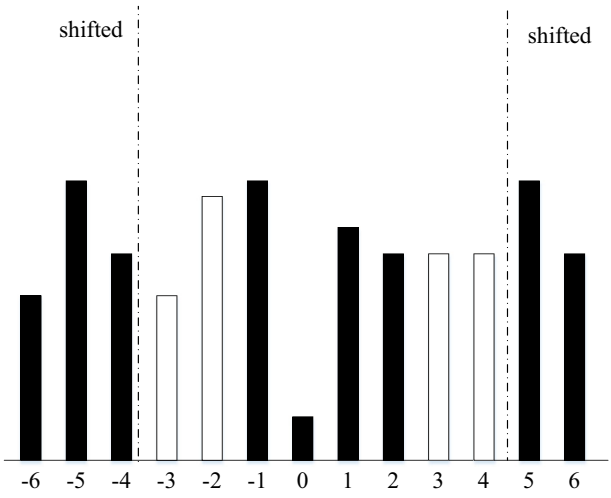

Fig. 3 The shifted histogram of *PE*

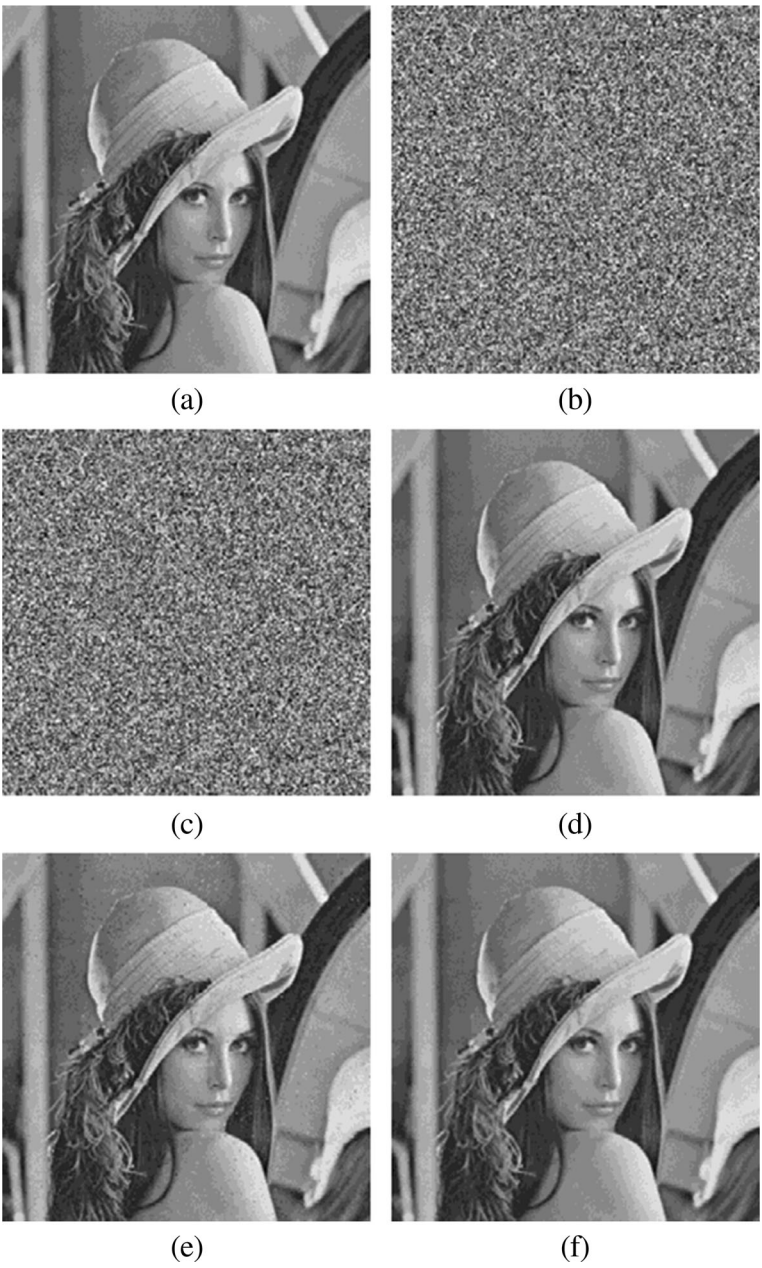

Fig. 4 (**a**) original AMBTC image Lena with compression ratio 0.625, (**b**) encrypted image, (**c**) cipher-image containing 5963 bits additional data, (**d**) recovered image, (**e**) directly decrypted image, (**f**) filtered version of (**e**) with PSNR 34.00 dB

Case 2: if $PE_i = -1$, $L_i = 1$ and corresponding location map is 1, then the bit plane should be replaced by binary additional data to be embedded and $flag_i = 1$;

Case 3: if $PE_i = 0$, $L_i \neq 1 and L_i \neq 254$ or corresponding location map is 0, then the bit plane should be replaced by binary additional data to be embedded and $flag_i = 1$;
Case 4: otherwise, $flag_i = 0$;

Then modify L_i according to eq. (11) to embed additional data.

However, this process raises the issue of how to embed auxiliary information so that it needs not to be sent to receiver via separate channel. Auxiliary information consists of two pairs of peak point and zero point (36 bits), the length of the location map ($\lceil \log_2 N \rceil$ bits), and the location map. The auxiliary information can be compressed and encrypted with AES etc. then it replaces the bit plane of the first several triples. Those triples will be skipped in the data embedding phase. The original bit plane of those triples will be encrypted with AES etc. and embedded into the remaining triples together with the additional data. This then solves all of the problems in the data embedding phase. Obviously, AMBTC image $\tilde{I}$, consisting of N triples, is processed triple by triple in additional data embedding and collecting of auxiliary information, so, the time complexity of the data embedding process is $O(N)$.

3.3 Data extraction and image recovery

When the receiver only has the data-hiding key, he can decrypt the auxiliary information, including two pairs of peak points and zero points (P_1 and Z_1, P_2 and Z_2), the length of the location map, and the location map itself. According to the decrypted auxiliary information, the data can then be extracted. The detail of the extraction phase is:

Step 1: for the i-th triple, calculate PE_i according to eq. (10), set $flag_i = 0$;
Step 2: recover L_i and extract additional data from the$i$-th triple, 6 cases are considered:
Case 1: if $Z_1 \leq PE_i < P_1 - 1$, then set $L_i = L_i^{'} - 1$;
Case 2: if $PE_i = P_1 - 1$, then set $d = 1$, and $L_i = L_i^{'} - 1$;
Case 3: if $PE_i = P_1$, then set $d = 0$, $L_i = L_i^{'}$, and $flag_i = 1$;
Case 4: if $PE_i = P_2$, then set $d = 0$, $L_i = L_i^{'}$, and $flag_i = 1$;
Case 5: if $PE_i = P_2 + 1$, then set $d = 1$, and $L_i = L_i^{'} + 1$;
Case 6: if $P_2 + 1 < PE_i \leq Z_2$, then set $L_i = L_i^{'} + 1$;

Where d is the extracted data, and $flag_i$ is merely used for recording whether $PE_i = P_1$ or $PE_i = P_2$;

Step 3: To extract additional data in bit plane, 3 cases are considered:
Case 1: if $PE_i = 1$, $L_i = 254$, and corresponding location map is 1, then the bits in the bit plane are additional data and one bit of data d extracted in step 2 should be abandoned if $flag_i = 1$;
Case 2: if $PE_i = -1$, $L_i = 1$, and corresponding location map is 1, then the bits in bit plane are additional data and one bit of data d extracted in step 2 should be abandoned if $flag_i = 1$;
Case 3: if $PE_i = 0$, $L_i \neq 1$ and $L_i \neq 254$ or corresponding location map is 0, then the bits in bit plane are additional data and one bit of data d extracted in step 2 should be abandoned if $flag_i = 1$;

Case 4: recover encrypted AMBTC image according to the location map. If $L_i = 1$ or $L_i = 254$, and corresponding location map is 1, then set $L_i = 0$ or $L_i = 255$, respectively.

By this stage, additional data have been extracted from the triples in the encrypted AMBTC-compressed image and decrypted with the data-hiding key. Note that the first several triples into which auxiliary information has been embedded should be skipped in data extraction phase.

When the receiver only has the image encryption key, he can generate pseudo-random binary numbers $r_{i,j}$ and $s_{i,k}$, get the bits of L_i and H_i according to eq. (3) and (4), then decrypt L_i, H_i, and B_i according to eq. (5), (6), and (9). Median filtering is adopted here to ensure the quality of decrypted image. After filtering, the directly decrypted image is similar to the original AMBTC-compressed image because only the LSB of L_i are modified in the embedding phase.

When the receiver has both the data-hiding and the image encryption keys, additional data can be extracted and the encrypted image can be recovered, after that, the AMBTC image can be decrypted exactly.

As for the efficiency of this stage, no matter which option is chosen, each triple in the AMBTC image is visited once, so its time complexity is also $O(N)$.

4 Experimental results

All experimental results are performed with MATLAB. Nine standard gray-scale images sized 512×512 and UCID datasets containing 1338 images (http://vision.cs.aston.ac.uk/datasets/UCID/ucid.html) are used here. Nine standard gray-scale images are: Lena, Sailboat, Peppers, Tiffany, Boat, Jet, Baboon, Splash and Man. We divide these images into 2×2,4×2,8×2,2×4,4×4,8×4,8×2,8×4,8×8 non-overlapping blocks to compress these images into AMBTC images.

Fig. 4(a) gives the original AMBTC image with compression ratio 0.625. After encryption, the original content of image is invisible as shown in Fig. 4(b). The marked cipher-image is shown in Fig. 4(c), which is the same as Fig. 4(b) when observed by the naked eye. If the

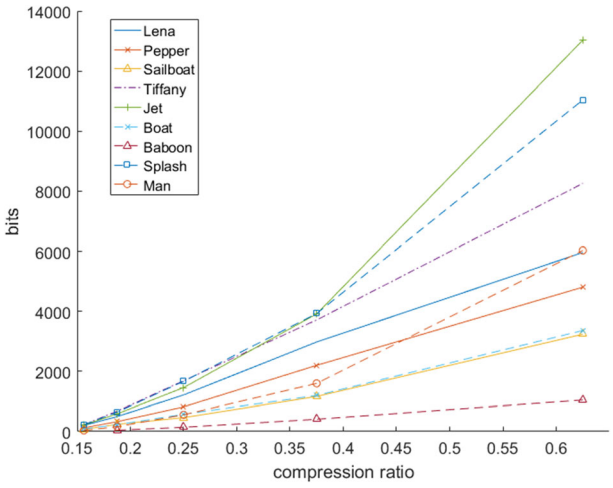

Fig. 5 The payload of different AMBTC images under different compression ratios

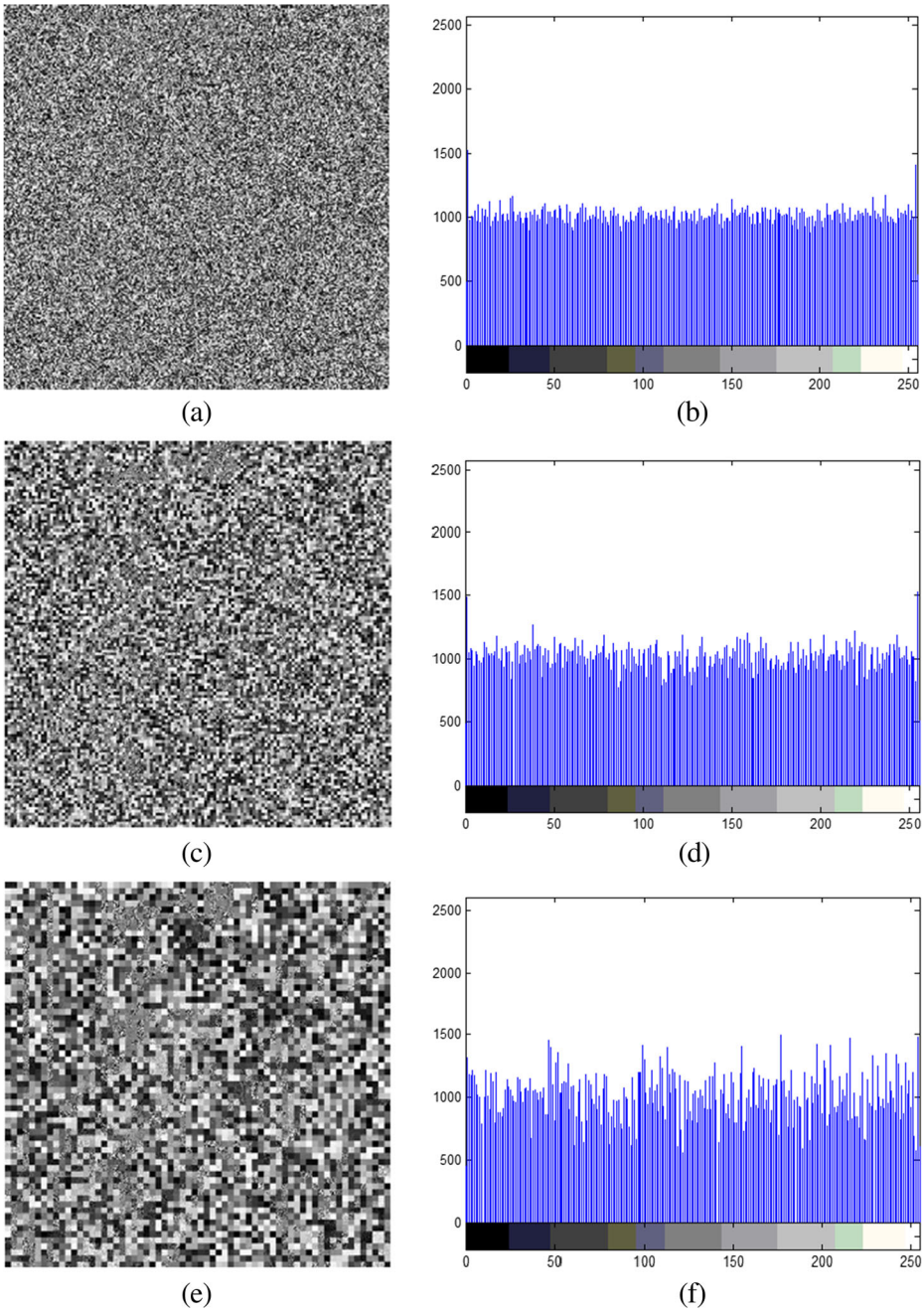

Fig. 6 The encrypted AMBTC image Lena under compression ratio (**a**) 0.625, (**c**) 0.25, (**e**) 0.156, and corresponding image histogram (**b**), (**d**), (**f**)

receiver has both the data-hiding and image encryption keys, the additional data can be extracted and Fig. 4(c) can be recovered error free, to get Fig. 4(d). However, if the receiver only has the image-encryption key, they can generate Fig. 4(e) by decrypting Fig. 4(c) directly

and filter noise by using median filtering, which is shown in Fig. 4(f). The PSNR of the directly decrypted image is 34.00 dB, and the distinction between the original AMBTC image and the directly decrypted image is also invisible to the naked eye.

Fig. 5 gives the payload of different AMBTC images under different compression ratios. The horizontal axis shows the compression ratio, and the vertical axis shows additional data embedded in AMBTC images. It can be observed that more additional data are embedded into those images with a higher compression ratio, which means an AMBTC image with high compression ratio provides more redundant space.

The results of encryption of AMBTC image Lena under different compression ratios are shown in Fig. 6. The higher the compression ratio is, the better the encryption result is. A low compression ratio implies that the block size adopted in the compression procedure is large, so the encryption results of those AMBTC images with low compression ratios are not as good as high ones.

Table 1 lists payload, PSNR of directly decrypted images (dB) and PSNR of recovered image (dB) for different AMBTC images when the block size in compression procedure is 2×2 and 4×4 respectively. The PSNR of directly decrypted images are above 27 dB in most cases. The image quality is better when the block size in the compression procedure is 2×2 than 4×4. The PSNR of the recovered image is infinite because the proposed method can recover the marked cipher-image without error when the receiver has both the data-hiding key and the image encryption key.

Fig. 7 gives the embedding rate of auxiliary information and additional data which is obtained by the amount of bits dividing the amount of image pixels when tested in 1338 UCID images. The horizontal axis is the number of images, and the vertical axis is corresponding average bits of auxiliary information and additional data embedded in each pixel. It is obvious that different images with the same block size 2×2 produce similar amount of auxiliary information because each image is encrypted with pseudo-random numbers. But the amount of additional data will vary with original images' redundancy. It means that the proposed method can embed a considerable number of bits in images with satisfactory encryption results.

5 Conclusion

In this paper, a reversible data hiding for encrypted AMBTC-compressed images is presented, which is divided into three phases: image encryption, data embedding, and data extraction and image recovery. In the data embedding phase, we encrypt the higher mean and lower mean of a

Table 1 Payload (bits), PSNR of directly decrypted images (dB) and PSNR of recovered images (dB) with different block size for different AMBTC images

	Payload		PSNR of directly decrypted images		PSNR of recovered images	
Block size	2×2	4×4	2×2	4×4	2×2	4×4
Lena	5963	1213	34.00	30.04	$+\infty$	$+\infty$
Pepper	4807	819	33.79	29.24	$+\infty$	$+\infty$
Sailboat	3323	462	29.81	27.27	$+\infty$	$+\infty$
Tiffany	8259	1687	33.18	27.46	$+\infty$	$+\infty$
Jet	13,028	1458	32.40	27.66	$+\infty$	$+\infty$
Boat	3360	554	30.32	28.78	$+\infty$	$+\infty$
Baboon	1047	141	23.66	23.40	$+\infty$	$+\infty$
Splash	11,043	1666	34.83	30.16	$+\infty$	$+\infty$
Man	6032	556	30.51	27.68	$+\infty$	$+\infty$

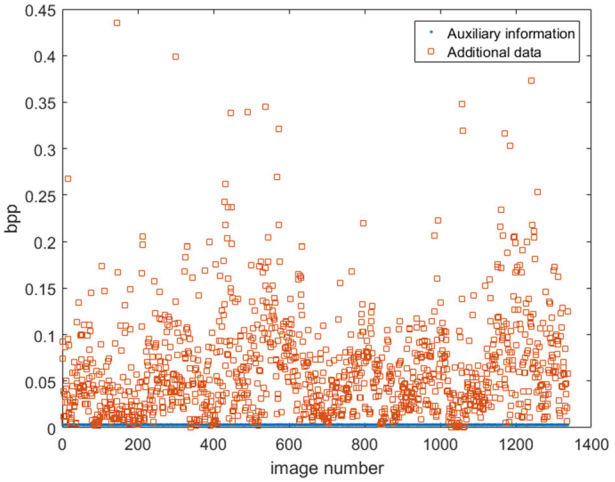

Fig. 7 The embedding rate of auxiliary information and additional data in 1338 UCID images with compression ratio 0.625

triple in an AMBTC-compressed image with the same random numbers, therefore, the correlation of higher mean and lower mean of nature image is preserved. The redundant space of the encrypted AMBTC-compressed image can then be exploited by using the histogram of prediction error in the absence of original image content in the data embedding phase. Experimental results and analysis demonstrate that, with the marked cipher-image, the receiver is able to exactly extract additional data by using the data hiding key, return a decrypted image very similar to the original image with the image encryption key, or extract additional data and recover the original image error free with both keys. In addition, the proposed method is very simple and its computational complexity is low so that it is applicable to real-time transmission.

Acknowledgements This research work is partly supported by the National Natural Science Foundation of China (61502009, 61525203, 61472235, U1636206), China Postdoctoral Science Foundation (2016 M591650), "Shu Guang" project supported by Shanghai Municipal Education Commission and Shanghai Education Development Foundation, Anhui Provincial Natural Science Foundation (1508085SQF216), Key Program for Excellent Young Talents in Colleges and Universities of Anhui Province (gxyqZD2016011) and Undergraduates Training Foundation of Anhui University (J10118511269).

References

1. Cao X, Du L, Wei X, Meng D, Guo X (2015) High capacity reversible data hiding in encrypted images by patch-level sparse representation. IEEE Trans on Cybernetics. doi:10.1109/TCYB.2015.2423678
2. Fu Z, Wu X, Guan C, Sun X, Ren K (2016) Toward Efficient multi-keyword fuzzy Search over encrypted Outsourced data with accuracy improvement. IEEE Trans Inf Forensics Secur 11(12):2706–2716
3. Fu Z, Ren K, Shu J, Sun X, Huang F (2016) Enabling personalized Search over encrypted Outsourced data with efficiency improvement. IEEE Trans Parallel Distrib Syst 27(9):2546–2559
4. Fu Z, Huang F, Sun X, Vasilakos AV, Yang CN Enabling semantic Search based on conceptual graphs over encrypted Outsourced data. IEEE Trans Serv Comput. doi:10.1109/TSC.2016.2622697
5. Fu Z, Sun X, Ji S, Xie G (2016) Towards Efficient content-aware Search over encrypted Outsourced data in cloud. Proceedings of the 35th Annual IEEE International Conference on Computer Communications (IEEE INFOCOM). doi:10.1109/INFOCOM.2016.7524606
6. Hong W, Chen TS (2012) A novel data embedding method using adaptive pixel pair matching. IEEE Trans Inf Forensics Secur 7(1):176–184

7. Hong W, Chen TS, Wu HY (2012) An improved reversible data hiding in encrypted images using side match. IEEE Signal Process Lett 19(4):199–202
8. Huang F, Huang J, Shi YQ (2016) New framework for reversible data hiding in encrypted domain. IEEE Trans Inf Forensics Secur 11(12):2777–2789
9. Lema MD, Mitchell OR (1984) Absolute moment block Truncation coding and its application to color image. IEEE Trans Commun 32(10):1148–1157
10. Li J, Li X, Yang B, Sun X (2015) Segmentation-based image Copy-move forgery detection Scheme. IEEE Trans Inf Forensics Secur 10(3):507–518
11. Ni Z, Shi YQ, Ansari N, Su W (2006) Reversible data hiding. IEEE Trans Circuits Syst Video Technol 16(3):354–362
12. Ou B, Li X, Zhang W (2015) PVO-based reversible data hiding for encrypted images. IEEE China Summit and International Conference on Signal and Information Processing (ChinaSIP). 831–835
13. Puech W, Chaumont M, Strauss O (2008) A reversible data hiding method for encrypted images. Security, Forensics, Steganography, and Watermarking of Multimedia Contents X, Proc. SPIE. 6819
14. Qian Z, Zhang X, Wang S (2014) Reversible data hiding in encrypted JPEG Bitstream. IEEE Trans Multimedia 16(5):1486–1491
15. Qian Z, Zhang X, Ren Y, Feng J (2016) Block cipher based Separable reversible data hiding in encrypted images. Multimedia Tools and Applications 75(21):13749–13763
16. Qian Z, Zhou H, Zhang X, Zhang W (2016) Separable reversible data hiding in encrypted JPEG bitstreams. IEEE Transactions on Dependable and Secure Computing
17. Qin C, Zhang X (2015) Effective reversible data hiding in encrypted image with privacy protection for image content. J Vis Commun Image Represent 31:154–164
18. Qin C, Chang CC, Lin CC (2015) An adaptive reversible Steganographic Scheme based on the just noticeable distortion. Multimedia Tools and Applications. 74(6):1983–1995
19. Qin C, Chang CC, Hsu TJ (2015) Reversible data hiding Scheme based on exploiting modification direction with two Steganographic images. Multimedia Tools and Applications. 74(15):5861–5872
20. Tian J (2003) Reversible data embedding using a difference expansion. IEEE Trans Circuits Syst Video Technol 13(8):890–896
21. Wang Z, Zhang X, Yin Z (2016) Hybrid distortion function for JPEG steganography. J Electron Imaging 25(5):050501
22. Wang J, Li T, Shi YQ, Lian S, Ye J Forensics feature analysis in quaternion wavelet domain for distinguishing photographic images and computer graphics. Multimedia Tools and Applications. doi:10.1007/s11042-016-4153-0
23. Xia Z, Wang X, Sun X, Wang B (2014) Steganalysis of least significant bit matching using multi-order differences. Security and Communication Networks 7(8):1283–1291
24. Xia Z, Wang X, Zhang L, Qin Z, Sun X, Ren K (2016) A privacy-preserving and Copy-deterrence content-based image retrieval Scheme in cloud Computing. IEEE Trans Inf Forensics Secur 11(11):2594–2608
25. Xia Z, Wang X, Sun X, Liu Q, Xiong N (2016) Steganalysis of LSB matching using differences between nonadjacent pixels. Multimedia Tools and Applications. 75(4):1947–1962
26. Xu D, Wang R Separable and error-free reversible data hiding in encrypted images. Signal Process. doi:10.1016/j.sigpro.2015.12.012
27. Yin Z, Abel A, Zhang X, Luo B (2016) Reversible Data Hiding in Encrypted Image Based on Block Histogram Shifting. In 2016 I.E. International Conference on Acoustics, Speech and Signal Processing (ICASSP). 2129–2133
28. Zhang X (2010) Efficient data hiding with plus-minus one or two. IEEE Signal Process Lett 17(7):635–638
29. Zhang X (2011) Reversible data hiding in encrypted image. IEEE Signal Process Lette 18(4):255–258
30. Zhang X (2013) Commutative reversible data hiding and encryption. Security and Communication Networks. 6:1396–1403
31. Zhang W, Ma K, Yu N (2013) Reversibility improved data hiding in encrypted images. Signal Process 94: 118–127
32. Zhang X, Long J, Wang Z, Cheng H Lossless and reversible data hiding in encrypted images with public key cryptography. IEEE Trans Circuits Syst Video Technol. doi:10.1109/TCSVT. 2015.2433194
33. Zhou Z, Wang Y, Wu QMJ, Yang CN, Sun X (2017) Effective and Efficient global context verification for image Copy detection. IEEE Trans Inf Forensics Secur 12(1):48–63

Zhaoxia Yin received her B.Sc., M.E. and Ph.D. from Anhui University in 2005, 2010 and 2014 respectively. She is an ACM member, associate chair of academic committee of CCF YOCSEF Hefei 2016-2017 and is currently working as an associate professor in Anhui University. Her primary research focus is on information hiding, and she has published many SCI/EI indexed papers in journals, edited books and refereed conferences.

Xuejing Niu was born in China in 1994 and she is a college student of Anhui University. She participated in Research Training Foundation for Student of Anhui University in 2014 and Undergraduates Training Foundation of Anhui University in 2016. Her research interests include computer vision, machine learning and information hiding.

Xinpeng Zhang received the B.S. degree in computational mathematics from Jilin University, China, in 1995, and the M.E. and Ph.D. degrees in communication and information system from Shanghai University, China, in 2001 and 2004, respectively. Since 2004, he has been with the faculty of the School of Communication and Information Engineering, Shanghai University, where he is currently a Professor. His research interests include information hiding, image processing and digital forensics. He has published over 200 papers in these areas.

Jin Tang was born in Anhui province of China in 1976. He received his BEng in Automation in 1999, and PhD in Computer Science in 2007 from Anhui University, Hefei, China. He has published some 100 papers in journals, edited books and refereed conferences. Some papers were published on the Pattern Recognition journal, CVPR, IJCAI, and AAAI conferences. He is at present a professor at Anhui University of China. His current research interests include image and graph matching, pattern recognition and computer vision.

Bin Luo was born in Anhui province of China in 1963. He received his BEng. Degree in electronics and MEng. degree in computer science from Anhui university of China in 1984 and 1991, respectively. From 1996 to 1997, he worked as a British Council visiting scholar at the University of York under the Sino-British Friendship Scholarship Scheme (SBFSS). In 2002, he was awarded the Ph.D. degree in Computer Science from the University of York, the United Kingdom. He joined the University of York as a research associate from 2000 to 2004. He was a short term research fellow of British Telecom in 2006. He served as a visiting fellow of the University of New South Wales, Australia in 2008. He was a TCT Exchange Fellow at the Nanyang Technological University, Singapore. He has published some 200 papers in journals, edited books and refereed conferences. Some papers were published on the IEEE T-PAMI, Pattern Recognition journal, CVPR, IJCAI, and AAAI conferences. He is at present a professor at Anhui University of China. His current research interests include graph spectral analysis, image and graph matching, statistical pattern recognition.