Multimed Tools Appl (2016) 75:13765–13778
DOI 10.1007/s11042-015-2920-y

Lossless data hiding algorithm for encrypted images with high capacity

Shuli Zheng[1] · Dandan Li[1] · Donghui Hu[1] · Dengpan Ye[2] · Lina Wang[2] · Jinwei Wang[3]

Received: 14 January 2015 / Revised: 12 August 2015 / Accepted: 25 August 2015 /
Published online: 12 September 2015

Abstract In this paper, a novel reversible data hiding algorithm for encrypted images is proposed. In encryption phase, chaotic sequence is applied to encrypt the original image. Then the least significant bits (LSBs) of pixels in encrypted image are losslessly compressed to leave place for secret data. With auxiliary bit stream, the lossless compression is realized by the Hamming distance calculation between the LSB stream and auxiliary stream. At receiving terminal, the operation is flexible, that is, it meets the requirement of separation. With the decryption key, a receiver can get access to the marked decrypted image which is similar to the original one. With data-hiding key, the receiver can successfully extract secret data from the marked encrypted image. With both keys, the receiver can get secret data and the exactly original image. Compared with existing methods, experiments show

✉ Shuli Zheng
ZSL251@163.com

Dandan Li
lidandan110506@126.com

Donghui Hu
hudh@hfut.edu.cn

Dengpan Ye
yedp2001@163.com

Lina Wang
lnawang@163.com

Jinwei Wang
wjwei_2004@163.com

[1] School of Computer and Information, Hefei University of Technology, Hefei 230009, China

[2] Computer School, Wuhan University, Wuhan 430072, China

[3] Jiangsu Network Monitoring Engineering Center, Nanjing University of Information Science and Technology, Nanjing 210044, China

the feasibility and efficiency of the proposed method, especially in aspect of embedding capacity, embedding quality and error-free recovery with increasing payload.

Keywords Information security · Reversible data hiding · Image encryption · Lossless compression

1 Introduction

Reversible data hiding (RDH) in images is a technique, by which the original image can be losslessly recovered after the extraction of the secret data. This important technique can be applied to many scenarios, such as law forensics, military imagery and medical imagery, where no distortion of the original image is allowed. For this reason, RDH has attracted considerable research interest.

Many reversible data hiding methods have been proposed recently. In literature [7, 8, 15], the differences between two adjacent pixels is expanded to generate a new LSB plane for accommodating secret data. In literature[10, 13, 14], spare space is saved for data embedding by shifting the histogram of cover data from its peak point towards its zero points. Another strategy for RDH is based on lossless compression, in which a data hider makes use of redundancy of the original cover to create a blank space for embedding the secret data [1, 12].

With the increasing demand of privacy protection, encryption [2, 3, 16] becomes an effective and popular means, which converts original image into unintelligible one. There are some applications while RDH can be applied for encrypted images. For instance [18], to protect the privacy of the patient, the medical images should have been encrypted, a database administrator may aim to embed the personal data into encrypted medical images. With the personal data, the database administrator can manage the images without knowing the original content. On the other hand, a doctor, having both the data-hiding and the encryption keys, can extract the personal data and recover the original content without any error. That means a RDH scheme for encrypted image is attractive.

Some attempts on RDH for encrypted images have been proposed. In literature [11], AES algorithm was applied to encrypt original image, and each block of an encrypted image can carry one secret bit by simple substitution, at the decryption stage, an analysis of local standard deviation of the marked encrypted image is used to extract the embedded data. However, the embedding capacity is low and the quality of image decrypted from the marked encrypted version is unsatisfactory. In literature [18], a data hider divided the encrypted image into several blocks, by flipping 3 LSBs of specific pixels in each block, one secret bit can be embedded into each block. The data extraction and image recovery proceed with the aid of spatial correlation in natural image. Hong et al. [6] improved the method in literature [18] by further exploiting the spatial correlation using a different estimation equation and side match technique to reduce the error rate. However, in literature [6, 18], encrypted image should be decrypted first before data extraction, that is, if a receiver has the data-hiding key only, he can not extract any data from the marked encrypted image. Aiming for separating data extraction from image decryption, Zhang [19] compressed the LSBs of encrypted image by finding the syndromes of a low-density parity check matrix to create an extra space for secret data. These methods achieve low embedding capacity or generate marked decrypted image with poor quality for high embedding capacity. Besides that, all of them are subject to some error rates on data extraction and /or image recovery

[6, 18, 19]. In [20], Zhang embedded the additional data into various bit planes by using a reversible manner, and a parameter optimization method is used to ensure a good payload-distortion performance. the data insertion/extraction can be performed in both the plain and encrypted domains.

Since it is difficult to losslessly vacate room from the encrypted images. Ma et al. [9] and Zhang [21] both emptied out room before image encryption, and achieved excellent performance. However, the operating of image encryption and data embedding is inseparable, which can not satisfy the application requirement of privacy protection.

This paper proposes a novel RDH for encrypted images based on lossless compression with high capacity. Not only does the proposed method separate data extraction from image decryption but also achieves excellent performance in two different prospects:

1. Reversibility is realized, that is, data extraction and image recovery are free of any error. Moreover, image encryption and data hidden process is separable in the proposed method, which meets the demand of digital image for privacy protection.
2. For given embedding rates, the PSNR of marked decrypted image is significantly improved; and for acceptable PSNR, the range of embedding rates is greatly enlarged.

The rest of the paper is organized as follows. The scheme of the proposed method is elaborated in Section 2. Experimental results with analysis and comparison are shown in Section 3. The paper is concluded in Section 4.

2 Proposed method

To enable data embedding in the encrypted image, we propose a novel method to significantly improve the performance in terms of the data embedding capacity and image quality of the marked decrypted image by using the Hamming distance to compress the encrypted image for data embedding. In this proposed scheme, we vacate room after encryption.

The procedure of reversible data hiding in encrypted images is composed of three steps: image encryption, data hiding, data extraction and image recovery, as illustrated in Fig. 1. The content owner first converts the original image **A** into its encrypted version **B** with encryption key r_0. Then the data hider compresses the LSBs of **B** with data-hiding key k_0 to reserve blank space for additional data, and the data hider produces marked encrypted image **C**. At the receiver side, since the data hiding only affects the LSBs, decryption without data extraction can result in a marked decrypted image similar to **A**. With k_0, the secret data

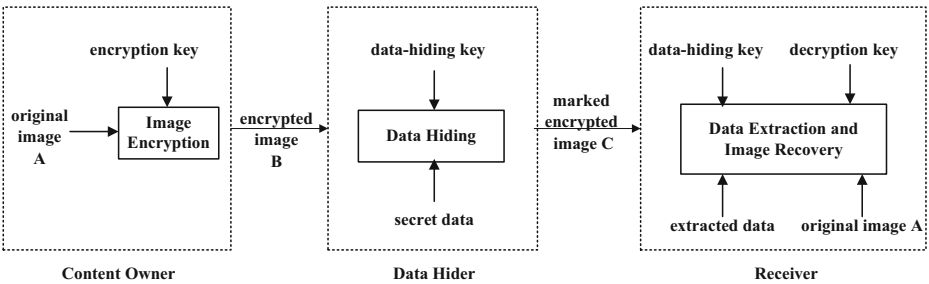


Fig. 1 The procedure of reversible data hiding for encrypted image

can be extracted successfully. With both of k_0 and r_0, the additional data can be correctly extracted and the original image can be exactly recovered.

2.1 Image encryption

Chaos [4] has several good properties including the ease of its generation, its sensitive dependence on initial condition and noise like, so that it is very suitable for image encryption. Meanwhile, it also indicate the trait of certainty, and can be completely reconstructed for image decryption as long as the system parameters and initial conditions are the same.

Logistic mapping is a classical model, which is used to studying the behavior of complex system of dynamic system, chaos, fractal ect. the encryption key as a initial value of Logistic mapping, a chaotic sequence is produced, and after the corresponding treatment, each element of the chaotic sequence is between 0 and 255.

Furthermore, the exclusive-or operation has reversibility. Combined with the Logistic mapping and the exclusive-or operation, a substitute encryption method is designed [5, 17].

In encryption phase, assume the original image **A** with a size of $N_1 \times N_2$ is in uncompressed format and each pixel with gray value falling into [0, 255] is represented by 8 bits. Denote the bits of pixel as $b_{i,j}(0), b_{i,j}(1), ..., b_{i,j}(7)$ where $1 \leq i \leq N_1, 1 \leq j \leq N_2$, the gray value as $p(i, j)$, and the number of pixels as N ($N = N_1 \times N_2$). Thus

$$b_{i,j}(k) = \left\lfloor \frac{p(i,j)}{2^k} \right\rfloor \bmod 2, 0 \leq k \leq 7 \tag{1}$$

The encrypted bits $b'_{i,j}(k)$ can be calculated through exclusive-or operation

$$b'_{i,j}(k) = b_{i,j}(k) \oplus r_{i,j}(k) \tag{2}$$

where $r_{i,j}(k)$ are created using the Logistic mapping with the encryption key r_0. Then, $b'_{i,j}(k)$ are concatenated orderly as the encrypted data.Chaos system can be used here to ensure that anyone without r_0, such as a potential attacker or a data hider, can not obtain any data about original content from the encrypted data.

2.2 Data hiding

In data hiding phase, once the data hider acquires the encrypted image **B**, he can embed secret data into it by modifying a proportion of encrypted data without having any information about the original image content, as shown in Fig. 2. The detailed procedure of data hiding is given as follows:

1. First, segment **B** into a number of non-overlapping blocks with the size of $l \times l$, l is a positive integer. Then, with the data-hiding key k_0, a random sequence is generated, and select t image blocks by using random interval method or random permutation method. t is a positive integer less than $\lfloor N/l^2 \rfloor$.
2. For each block B_i ($1 \leq i \leq t$), extract m LSBs from per pixel to generate a sequence $X_i = \{x_{i1}, x_{i2}, ..., x_{in}\}$, $n = (m \times l \times l)/3$, x_{ij} is composed of each three consecutive bits of X_i, $j \in [1, n]$. Then a corresponding sequence $Y_i = \{y_{i1}, y_{i2}, ..., y_{in}\}$ is generated, which should meet the condition that Hamming distance between x_{ij} and y_{ij} is no more than 1. Here Y_i is introduced as auxiliary bit sequence. For example, if $X_i = \{000, 001, 010, 100\}$, a corresponding sequence $Y_i = \{001, 001, 011, 100\}$, that is to say, if $x_{ij} = \{000\}$, a corresponding stream $y_{ij} = \{000\}$. m is a small positive integer less than 4.

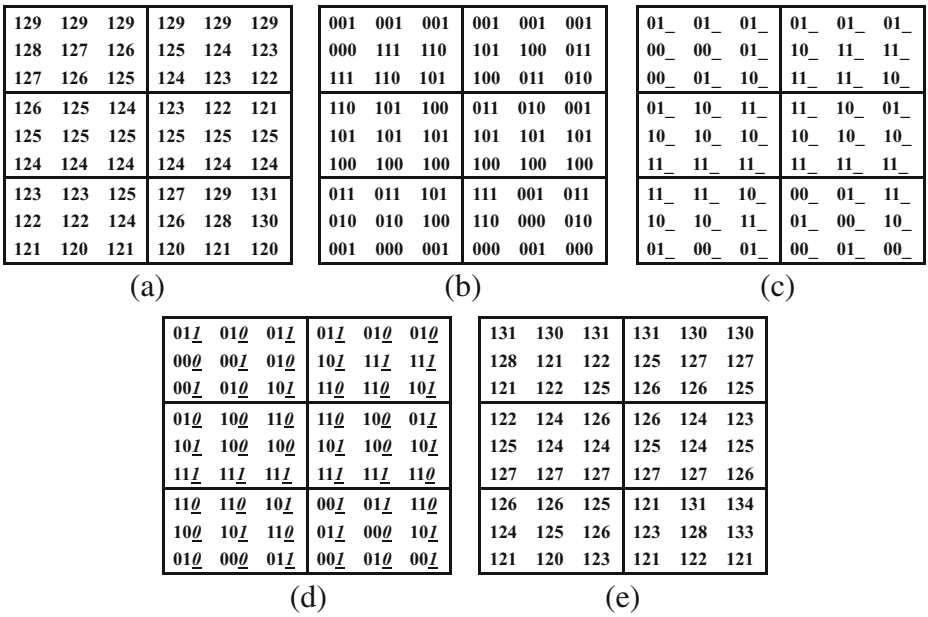


Fig. 2 The sample of proposed method (l=3,m=3, the shadows represent secret bits).(**a**) the encrypted image, (**b**) the 3 LSBs of each pixel, (**c**) the 2 LSBs after compression, (**d**) the 3 LSBs containing secret bit, (**e**) the marked encrypted image

3. The following (3) defines a set, denoted by $Coset1-4$, in which a stream containing two bits can replace two types of stream, each of them including three bits and the Hamming distance between them is three. On the basis of $Coset1-4$, x_{ij} can be compressed as two bits, so the third bit of x_{ij} is to leave for secret bit. Repeat this way, n bits are embedded into each block at most. For example, if $x_{ij}=\{000\}$, the compressed x_{ij} is $\{00\_\}$, shown in Fig. 2c.
4. Repeat above process until all the secert bits are embedded, a marked encrypted image is generated, denoted by **C**.

$$\begin{array}{cc} \underset{(00)}{Coset1} = \left\{\begin{array}{c}000\\111\end{array}\right\} & \underset{(01)}{Coset2} = \left\{\begin{array}{c}001\\110\end{array}\right\} \\ \underset{(10)}{Coset3} = \left\{\begin{array}{c}010\\101\end{array}\right\} & \underset{(11)}{Coset4} = \left\{\begin{array}{c}100\\011\end{array}\right\} \end{array} \tag{3}$$

In practice, $Coset1-4$ plays an important role in data embedment and image recovery process. Since the total $t \times n$ bits can be accommodated in all blocks at best, the maximum embedding rate, a ratio between the bit amount of secret data and the total number of cover pixels, is (4).

$$R_{\max} = \frac{t \times n}{N} = \frac{(N/(l \times l)) \times ((m \times l \times l)/3)}{N} = \frac{m}{3} \tag{4}$$

With the data-hiding key k_0 , the data hider randomly selects the blocks and makes proper extraction, thus anyone who does not possess the data-hiding key could not extract secret data.

2.3 Data extraction and image recovery

In this phase, since data extraction is completely independent from image decryption, there are three cases that a receiver has only the data-hiding key k_0, only the decryption key, here r_0, and both k_0 and r_0, respectively, demonstrated in Fig. 3.

Case 1 Data extraction from **C**

To manage and update personal information and ensure clients' privacy, a receiver may only get access to k_0. The operation of data extraction before image decryption ensures the feasibility of our work in this case. The receiver should do following three steps:

1. Divide the received image into blocks with the size of $l \times l$,according to the same way in data hiding phase.
2. With k_0, select blocks for data extraction by exploiting random interval method or random permutation method.
3. For each block, extract m LSBs from all pixels, generate a sequence $X'_i = \{x'_{i1}, x'_{i2}, ..., x'_{in}\}$ which is the same as the generation of X_i in data hiding phase and extract bit from the third bit of x'_{ij} successively. Finally, concatenate the extracted bits to form the secret bits. The whole process avoids the leakage of original image content.
4. Repeat the third step mentioned above, until all the secret data is extracted.

Furthermore, because each encrypted block is randomly selected and recombined, which makes any malicious attackers cannot obtain the embedded data under the absence of k_0.

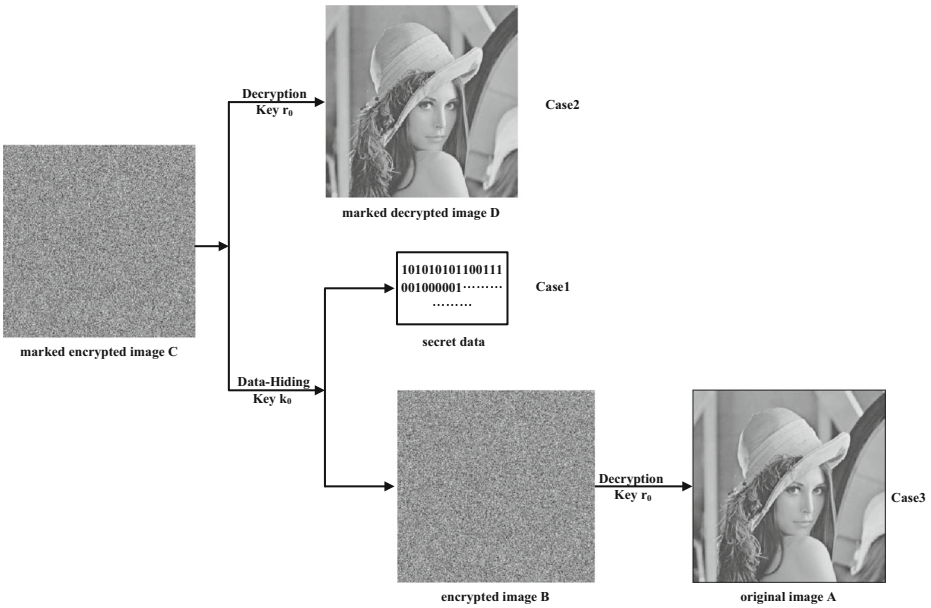


Fig. 3 Three cases in receiving terminal

Case 2 Generating the marked decrypted image **D**

Consider the case that the receiver has only r_0, he can generate $r_{i,j}(k)$ by Logistic mapping and r_0, and calculate the exclusive-or $r_{i,j}(k)$ and received data to decrypt the received image. The gray values of marked decrypted pixels $p'''_{i,j}(k)$ can be calculated by (5).

$$p'''(i,j) = \sum_{k=0}^{7} b'''_{i,j}(k) \cdot 2^k \tag{5}$$

where $b'''_{i,j}(k)$ are the binary bits of **D**.

Since the $(8-m)$ most significant bits (MSBs) of **D** are kept unchanged in data hidden process. Clearly, the original $(8-m)$ MSBs are retrieved correctly, the content of **D** is similar to the original image. However, the receiver is unable to obtain the secret data.

Case 3 Data extraction and image restoration

With the marked encrypted image **C**, if the receiver has both k_0 and r_0, he can further extract the secret data and recover the original image.

The data extraction process is essentially identical to Case 1, and the receiver can extract the secrte data successfully. After extracting the secret data, the receiver need to recover the original image, the following outlines the specific steps:

1. After data extraction, segment the remaining bits of X'_i into a number of streams x''_{ij} successively, x''_{ij} containing two bits. Then, find the corresponding streams $x(i,1)$ and $x(i,2)$ in $Coset1-4$, and the Hamming distance between $x(i,1)$ and $x(i,2)$ is three.

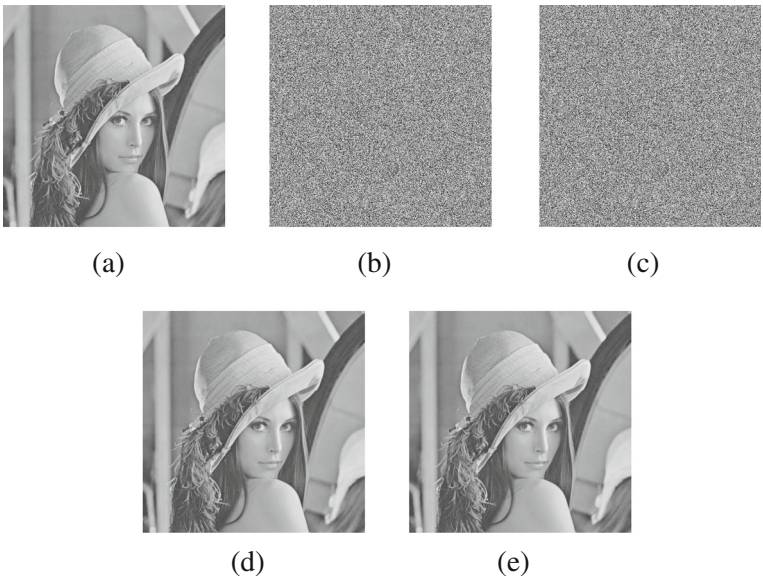


Fig. 4 (**a**) Original Lena, (**b**) its encrypted image, (**c**) marked encrypted image, (**d**) marked decrypted image, (**e**) recovery image

2. According to the corresponding sequence y_{ij}, the receiver need to calculate the Hamming distance between y_{ij} and $x\ (i, 1)$, $x\ (i, 2)$, respectively.
3. If the Hamming distance between $x\ (i, 1)$ and y_{ij} is no more than 1, $x\ (i, 1)$ is the bits of encrypted pixels; Similarly, if the Hamming distance between $x\ (i, 2)$ and y_{ij} is less than or equal to 1, $x\ (i, 2)$ is the bits of encrypted pixels.
4. Continue doing Step1 to Step3 and merger all encrypted bits to form the encrypted image.
5. With the encrypted image and r_0, obtain the original pixels via (2) and (5).

3 Experiments and comparisons

The proposed method will be tested on a number of standard images, which include Baboon, Boat, Barbara, Lake, Man and Lena. The size of all images is 512×512. The criteria PSNR is used to evaluate the quality of marked decrypted image. Moreover, the embedding rate is employed to evaluate the amount of the additional data. Note that in this section, the listed PSNR is the PSNR of the marked decrypted image versus the original image.

The standard image Lena shown in Fig. 4a is used as the original image in the experiment. Figure 4b is the encrypted image. Then, we let $m = 1$, $l = 40$, and embed 1.1×10^4 additional bits into the encrypted image, the marked encrypted image is given as Fig. 4c, and the embedding rate is 0.041 bit per pixel (bpp). With the marked encrypted image, we can generate the marked decrypted image using the encryption key, the PSNR of the marked decrypted image is $55.00dB$. The marked decrypted image is illustrated in Fig. 4d. Figure 4e depicts the recovery image which is identical to the original image. By both using the data-hiding and encryption keys, the embedded data can be successfully extracted and the original image can be recovered from the marked encrypted image.

Without loss of generality, the use of multiple LSBs takes into account the fact that the values of PSNR can be reduced with an increase in embedding rates. For different standard images, the comparison results measured by PSNR for three different choices of LSBs, i.e. the number of LSBs m, are presented in Table 1, where the embedding rate is measured by bits per pixel (bpp). It is observed that when embedding rate is small, one or two LSBs can be introduced. With a growing embedding rate, we prefer using three LSBs to single one. In practice, we can flexibility choose m in the light of the length of secret bits for acceptable PSNR.

In the proposed scheme, the higher the embedding rate, the worse the quality of the marked decrypted image, therefore, we should not only improve the embedding capacity but also ensure good quality of the marked decrypted image. The quality of marked decrypted images is compared in the term of PSNR. Figure 5 plots the PSNR of different marked decrypted images under given embedding rates. We modify the embedding way and embedding positions of the secret data: we randomly select a certain number of encrypted blocks with the size of 40×40. In the following, according to $Coset1 - 4$, every 3 LSBs of encrypted pixels are compressed as two bits, and a blank space is generated for secret data. At the same time, 5 MSBs of each encrypted pixel are kept unchanged. We have compared the proposed method with the state-of-the-art work [6, 18, 19]. In Fig. 5, the X-coordinate represents the embedding rate R, the Y-coordinate is the PSNR of the marked decrypted images. Take standard image Baboon for instance, the PSNR of the marked decrypted image using our method is about 15.36 dB higher than the methods in [6, 18, 19]. From Fig. 5, it can be observed that all range of the embedding rates, the PSNR of marked decrypted

Table 1 PSNR comparison for three different the number of LSBs m choice under various embedding rates

PSNR(dB)									
Embedding rate(bpp)		0.025	0.030	0.040	0.050	0.060	0.070	0.080	0.090
Baboon	m=1	58.18	56.71	51.17	45.96	44.28	40.01	38.86	37.67
	m=2	57.26	54.23	50.29	45.53	43.40	39.14	38.57	36.34
	m=3	54.35	53.62	49.10	44.53	42.51	39.08	37.80	35.12
Boat	m=1	60.18	59.13	56.67	52.01	49.17	46.74	41.10	40.47
	m=2	58.83	57.36	55.78	51.15	48.98	46.05	40.56	40.32
	m=3	57.64	56.24	54.05	50.00	47.83	45.86	40.03	39.48
Barbara	m=1	57.40	56.06	51.19	46.17	41.18	39.56	37.26	36.04
	m=2	56.12	55.79	50.62	45.53	40.62	38.30	36.78	34.06
	m=3	55.02	54.35	49.96	44.81	40.01	37.58	35.02	–
Lake	m=1	61.35	57.18	52.87	47.57	46.98	44.12	41.28	37.15
	m=2	60.03	56.83	52.16	46.85	46.03	43.72	40.72	36.45
	m=3	58.60	55.10	50.00	46.76	45.03	43.09	40.00	36.10
Man	m=1	65.84	62.72	58.62	55.66	55.58	52.53	49.84	47.59
	m=2	64.36	61.83	57.67	55.06	54.68	52.25	49.03	46.32
	m=3	63.54	60.32	56.79	54.03	53.76	52.05	48.98	44.06
Lena	m=1	67.06	61.23	58.35	54.88	52.08	51.16	50.45	46.18
	m=2	66.50	60.32	56.68	54.32	51.38	50.28	50.08	45.46
	m=3	64.69	59.08	55.68	53.37	50.07	49.93	49.93	45.25

image is improved about 15 dB than the methods in [6, 18, 19]. The gain in terms of PSNR is significantly higher at same embedding rate than state-of-the-art reversible data hiding algorithms [6, 18, 19]. In addition, another advantage of the proposed method is the much wider range of embedding rate for acceptable PSNRs. In fact, the proposed method can embed more than 5 times as large capacity for the same acceptable PSNR, e.g., PSNR=45dB, as the methods in [6, 18, 19].

In [6, 18], schemes of reversible data hiding for encrypted images are proposed. [6] made improvement on the basis of the work in [18] in order to achieve much lower error rate. these two methods only accommodated one secret bit in each block. However, our proposed method can embed one secret bit into each pixel at most. In [19], the least significant bits of the encrypted image are compressed to create a sparse space to accommodate the additional data, the embedding rate is relatively low, and with the decrease of block size, the original image can not be recovered perfectly.

As mentioned in Section 1, all methods in [6, 18, 19] maybe introduce some error on data extraction and/or image restoration, while the proposed method is free of any error for a number of images.

Our algorithm not only can hide a large number of the secret data, but also can recover the original image losslessly. Whats more, the operation sequence in receiving terminal is flexible, meanwhile it meets the requirement of privacy protection. All these make the proposed algorithm has better competitiveness.

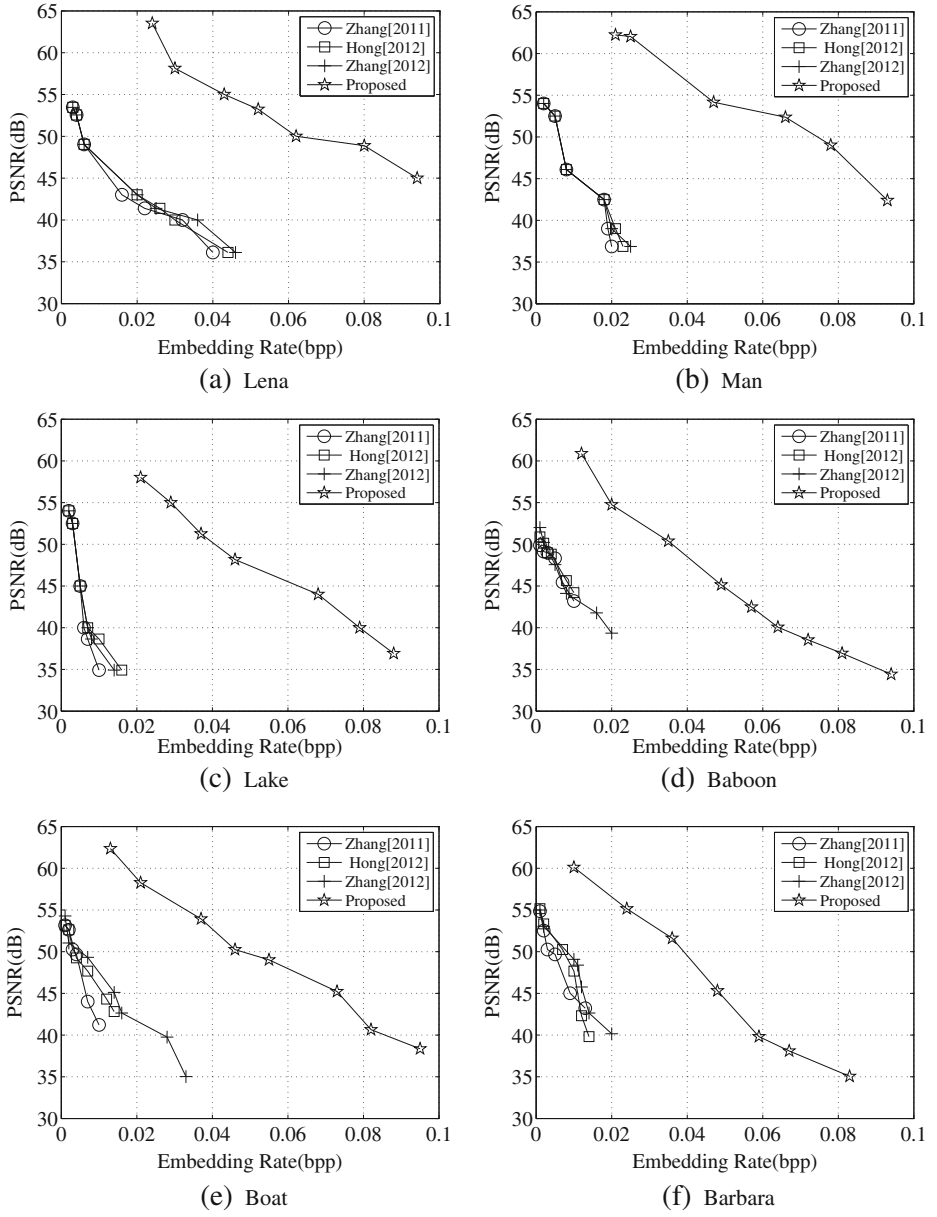


Fig. 5 Performance comparison of different algorithms for PSNR under given embedding rates

4 Conclusion

A novel scheme for reversible data hiding for encrypted images based on lossless compression with a high capacity is presented, which consists of image encryption, data hiding, and data extraction/image recovery phases. In encryption phase, a chaotic sequence is used to encrypt the original bits. In data hiding phase, although the data hider does not get access

to the original image, he makes use of the Hamming distance between streams to generate a blank space for secret data. With marked encrypted image, since the data hiding only affects LSBs, a decrypted version with the decryption key is similar to the original image. According to the data-hiding key, secret data can be correctly extracted. When using both of the data-hiding and the decryption keys, the secret data can be correctly extracted and the original image can be restored perfectly. Experimental results show that the proposed method can achieve reversibility, separate data extraction from image decryption, and effectively improvement on the embedding rate and the quality of marked decrypted images. In this work, one secret bit can be embedded into one encrypted pixel at most.

Acknowledgments This work is supported by the National Natural Science Foundation of China (Grant Nos. 61272540,61272453 and 61232016) , the Anhui Provincial Natural Science Foundation(Grant No. 1508085MF115) and the Priority Academic Program Development of Jiangsu Higher Education Institutions (PAPD).

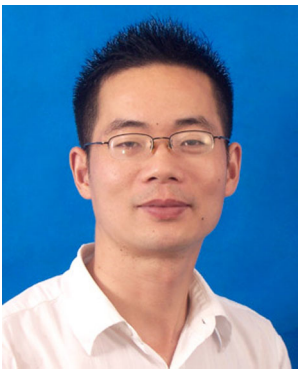
References

1. Celik MU, Sharma G, Tekalp AM, Saber E (2005) Lossless generalized-lsb data embedding. IEEE Trans Image Process 14:253–266
2. Chen G, Zhao X, Li J (2011) A self-adaptive algorithm on image encryption. J Softw 16:1975–1982
3. Fu Z, Sun X, Liu Q, Zhou L, Shu J (2015) Achieving efficient cloud search services: multi-keyword ranked search over encrypted cloud data supporting Parallel Computing. IEICE Transactions on Communications E98-B(1):190–200
4. Gao H, Zhang Y, Liang S, Li D (2006) A new chaotic algorithm for image encryption. Chaos, Solitons and Fractals 29:393–399
5. Gu Q, Yao M (2003) A research of digital image encryption based on logistic chaotic sequence. Computer Engineering and Applications 39:114–116
6. Hong W, Chen T-S, Wu H-Y (2012) An improved reversible data hiding in encrypted images using side match. IEEE Signal Proc Letters 19:199–202
7. Hu Y, Lee H-K, Li J (2009) De-based reversible data hiding with improved overflow location map. IEEE Trans Circuits and Systems for Video Technology 19:250–260
8. Joong KH, Sachnev V, Qing SY, Jeho N, Nam J, Choo H-G (2008) A novel difference expansion transform for reversible data embedding. IEEE Trans Information Forensics and Security 3:456–465
9. Ma K, Zhang W, Zhao X (2013) Reversible data hiding in encrypted images by reserving room before encryption. IEEE Trans Information Forensics and Security 8:553–562
10. Ni Z, Shi Y-Q, Ansari N, Su W (2006) Reversible data hiding. IEEE Trans Circuits and Systems for Video Technology 16:354–362
11. P W C M, S O (2008) A reversible data hiding method for encrypted images, International Society for Optics and Photonics, San Jose, CA, USA, 68191E-68191E-9
12. Qin C, Chang CC, Chen YC (2013) Efficient reversible data hiding for VQ-compressed images based on index mapping mechanism. Signal Process 93:2687–2695
13. Qin C, Chang CC, Huang YH et al. (2013) An inpainting-assisted reversible steganographic scheme using a histogram shifting mechanism. IEEE Trans Circuits and Systems for Video Technology 23:1109–1118
14. Tai W-L, Yeh C-M, Chang C-C (2009) Reversible data hiding based on histogram modification of puxel differences. IEEE Trans Circuits and Systems for Video Technology 19:906–910
15. Tian J (2003) Reversible data embedding using a difference expansion. IEEE Trans Circuits and Systems for Video Technology 13:890–896
16. Xia Z, Wang X, Sun X, Qian Wang (2015) A Secure and Dynamic Multi-keyword Ranked Search Scheme over Encrypted Cloud Data. IEEE Transactions on Parallel and Distributed Systems. doi:10.1109/TPDS.2015.2401003
17. Zhang L, Liao X, Wang X (2005) An image encryption approach based on chaotic maps. Chaos, Solitons and Fractals 24:759–765
18. Zhang X (2011) Reversible data hiding in encrypted image. IEEE Signal Proc Letters 18:255–258

19. Zhang X (2012) Separable reversible data hiding in encrypted image. IEEE Trans Information Forensics and Security 7:826–832
20. Zhang X (2013) Commutative reversible data hiding and encryption. Security and Communication Networks 6:1396–1403
21. Zhang W, Ma K, Yu N (2014) Reversiblility improve data hiding in encrypted images. Signal Process 94:118–127

Shuli Zheng received the B.S. degree in electric automatization from Hefei University of Technology in 1996, the Ph.D. degree in Computer Application of Hefei University of Technology in 2003. She is an associate professor at the School of Computer and Information, Hefei University of Technology. Her research interests include information hiding, wireless network security and multimedia content security.

Dandan Li received the B.S. degree in school of information from the Anhui Xinhua University in 2012. She is currently a M.S. student in the School of Computer and Information, Hefei University of Technology. Her research interests include reversible data hiding and multimedia content security.



Donghui Hu received the B.S. degree in mathematics from the Anhui Normal University in 1995, the M.S. degree in computer science and technology from University of Science and Technology of China in 2004, and the Ph.D. degree in information security from Wuhan University in 2010. He is currently an associate professor in the School of Computer and Information, Hefei University of Technology. His research interests include information security, digital image forensic and steganalysis, and privacy protection.

Dengpan Ye received the B.S. degree in automatic control from South China University of Technology in 1996, the Ph.D. degree in automatic control from NanJing University of Science and Technology in 2005. He worked as a Post-Doctoral Fellow in Information System School of Singapore Management University. And since 2012 he has been a professor in the Computer School of Wuhan University. His research interests include multimedia information hiding against, video watermarking, perceptual hashing, multimedia forensics, and information security.

Lina Wang received the M.S. degree in computer science and technology from Northeastern University in 1989, and the Ph.D. degree in computer science and technology from Northeastern University in 2001. Now she works as a professor in the Computer School of Wuhan University. Her research interests include information hiding, digital watermark, and network security.

Jinwei Wang received the B.S. degree from Inner Mongolia Electric Power College in 2000, the Ph.D. degree in information security from NanJing University of Science & Technology in 2007. Now he works as an associate professor at Nanjing University of Information Science and Technology. His research interests include multimedia watermarking, multimedia forensics and multimedia and multimedia encryption.