Complete Separable Reversible Data Hiding in Encrypted Image

Yin Zhaoxia[1,2], Wang Huabin[1], Zhao Haifeng[1], Luo Bin[1(✉)], and Zhang Xinpeng[2]

[1] Key Laboratory of Intelligent Computing and Signal Processing, Ministry of Education, Anhui University, Hefei 230601 People's Republic of China
{yinzhaoxia, wanghuabin, senith}@ahu.edu.cn, Luobin_ahu@163.com

[2] School of Communication and Information Engineering, Shanghai University, Shanghai 200072, People's Republic of China
xzhang@shu.edu.cn

Abstract. Reversible data hiding in encrypted image (RDHEI) is an emerging technology since it has good potential for practical applications such as encrypted image authentication, content owner identification and privacy protection. But there is one key problem of many existing published works, that the embedded data only can be extracted either before or after image decryption. In this paper, a complete separable reversible data hiding scheme in encrypted images is proposed. Additional data can be embedded into a cipher image which is encrypted by RC4 and can be extracted error-free both from the cipher domain and the plaintext domain. Moreover, the proposed method is simpler to calculate, while offering better performance. The results demonstrate that larger payload, better image quality, and error-free data extraction as well as image recovery are achieved.

Keywords: Reversible data hiding in encrypted images (RDHEI) · Privacy protection · Histogram modification

1 Introduction

Data hiding refers to technology that is used to embed additional data into multimedia and can be divided into non-reversible [1, 2] and reversible categories [3–10]. Reversible data hiding can be achieved mainly based on lossless compression [3], integer transform [4], difference expansion (DE) [5] and histogram shifting (HS) [6–8]. All of these methods have good embedding efficiency for plaintext images and can also be applied to JPEG images [9, 10].

As a typical SPED (signal processing in the encrypted domain [11]) topic, RDHEI means embedding additional data into encrypted images, and has the reversibility feature of being able to extract the additional data and recover the original image. Since there is good potential for practical applications including encrypted image authentication, content owner identification, and privacy protection, RDHEI has attracted more and more attention from many researchers [12–20].

Z. Huang et al. (Eds.): ICCCS 2015, LNCS 9483, pp. 101–110, 2015.
DOI: 10.1007/978-3-319-27051-7_9

In [15], an image is encrypted by a stream cipher and the data hider can embed additional data by flipping the 3 LSB (least significant bits) of pixels. Hong et al. [16] improve on this with side block matching and smoothness sorting. This year, Liao and Shu proposed an improved method [17] based on [15, 16]. A new, more precise function was presented to estimate the complexity of each image block and increase the correctness of data extraction/image recovery. However, in all of the methods mentioned above [15–17], data can only be extracted after image decryption. To overcome this problem, a separable RDHEI is proposed [18]. A legal receiver can choose 3 different options depending on the different keys held: extracting only the embedded data with the data hiding key, decrypting an image very similar to the original with the content owner key, or extracting both the embedded data and recovering the original image with both of the keys. Recently, another separable method based on pixel prediction was proposed in [19]. In the data hiding phase, a number of individual pixels are selected using a pseudo-random key, and additional bits are hidden in the two most significant bits. However, as the payload increases, the error rate also increases. Yin et al. [20] offer high payload and error-free data extraction by introducing multi-granularity permutation, which does not change the image histogram. However, leakage of the image histogram is inevitable under exhaustive attack. Moreover, in all of the methods discussed above [18–20], the embedded data can only be extracted before image decryption. That means that a legal receiver who has the data hiding key and the decrypted image cannot extract the embedded data.

To solve this problem, this paper presents a new complete separable RDHEI method based on RC4 encryption [21] and local histogram modification. Not only can the proposed method completely satisfy the definition of "separable" [18], but the embedded data can be extracted error-free both from marked encrypted images (cipher domain) and directly decrypted images (plaintext domain). However, there is a tradeoff: there should be no saturation pixels of value 0 or 255 in the image. Since saturation pixels almost are non-existent in natural images, this is a small concession. Compared with other state-of-the-art research [18, 19], the proposed method achieves higher embedding payload, better image quality and error-free image restoration.

2 Proposed Method

The framework of the proposed method is shown in Fig. 1. An image I can be encrypted to produce the encryption version I_e by using RC4 image encryption approach. This is a symmetric cipher technology and the decryption key is the same as the encryption key. Here, for simplicity, we shall call it the content owner key K_c. With the data hiding key K_d, the data hider can embed additional data A into encrypted image I_e and the marked encrypted image I'_e is generated.

On the receiver side, data extraction is completely independent from image decryption. The embedded data can be extracted from the decrypted version I' after image decryption, and also can be extracted from the cipher domain I'_e directly. With both of the keys K_c and K_d, the original image I can be reconstructed error-free. The details of image encryption, data embedding, data extraction and image recovery are

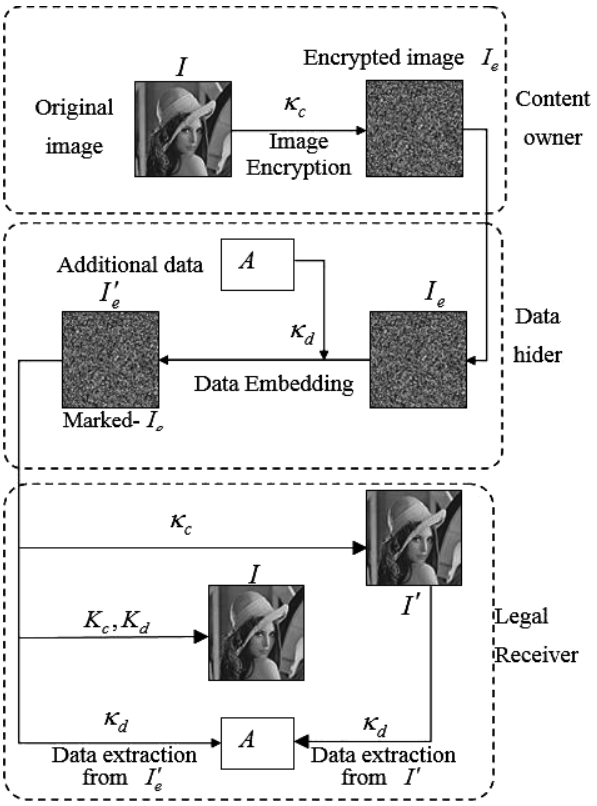

Fig. 1. Framework of the proposed method

elaborated in the following sections. First, we discuss image encryption and decryption by using RC4 [21] in Sect. 2.1.

2.1 Image Encryption and Decryption

Given a gray image $I = \{p_i\}_{i=1}^{n}$ sized n pixels containing no saturation pixels, p_i is the value of the i-th pixel and $p_i \in \{1, 2, \ldots, 253, 254\}$. We choose $K = \{k_j\}_{j=1}^{l}$, a randomly generated key-stream sized l using RC4 from a secret seed S_k. Then the image encryption is performed pixel by pixel as given in Eqs. (1) and (2) to get the encrypted image $I_e = \{q_i\}_{i=1}^{n}$:

$$I_e = e(I, K) \tag{1}$$

$$\begin{aligned} e(I, K) &= (I + K) \bmod 254 + 1 \\ &= \{(p_i + k_j) \bmod 254 + 1\}_{i=1}^{n} \\ &= \{q_i\}_{i=1}^{n} \end{aligned} \tag{2}$$

The decryption:

$$I = d(I_e, K) \tag{3}$$

$$d(I_e, K) = (I_e - 1 - K) \bmod 254 \tag{4}$$

Since the value of each grayscale pixel p_i ranges from 1 to 254, Eq. (4) has a unique solution. If the solution equals to 0, it can be revised to 254. In this paper, we divide the original image into non-overlapping blocks $I = \{B_j\}_{j=1}^{l}$ sized $u \times v$ at first, where $l = n/(u \times v)$. Then all the pixels in each block can be encrypted with the same k_j. Thus, each encrypted block B_j^e keeps structure redundancy to carry additional data.

2.2 Data Embedding

After image encryption, additional data can be embedded into each cipher block of $I_e = \{B_j^e\}_{j=1}^{l}$ to generate marked version $I_e^{'} = \{B_j^{'e}\}_{j=1}^{l}$ based on local histogram modification, which will be described in detail in this section.

Firstly, two pixels of each block are selected randomly to use as the basis pixels, and the basis pixel values are kept unchanged during data embedding.

To carry out this process, for each image block $\{B_j^e\}_{j=1}^{l}$ sized $u \times v$, the two basic pixels are denoted by $\hat{q}_{j,L}$, $\hat{q}_{j,R}$ and the remaining $u \times v - 2$ pixels are denoted by $\{\bar{q}_{j,k}\}_{k=1}^{u\times v-2}$, i.e. $B_j^e = \{\hat{q}_{j,L}, \hat{q}_{j,R}, \bar{q}_{j,k}\}_{k=1}^{u\times v-2}$. Using the basis pixels $\hat{q}_{j,L}$, $\hat{q}_{j,R}$, two peaks in each block are determined, with $g_{j,L}$ and $g_{j,R}$ identified as Eqs. (5) and (6):

$$g_{j,L} = \min(\hat{q}_{j,L}, \hat{q}_{j,R}) \tag{5}$$

$$g_{j,R} = \max(\hat{q}_{j,L}, \hat{q}_{j,R}) \tag{6}$$

The data hider then scans the non-basic pixels $\{\{\bar{q}_{j,k}\}_{k=1}^{u\times v-2}\}_{j=1}^{l}$ (i.e. excluding the two basis pixels used to determine peak values) to conceal the additional data A.

To do this, if a scanned pixel $\bar{q}_{j,k}$ is equal to the value of $g_{j,L}$ or $g_{j,R}$, a bit x extracted from A is embedded by modifying $\bar{q}_{j,k}$ to $q'_{j,k}$ according to Eq. (7).

$$q'_{j,k} = \begin{cases} \bar{q}_{j,k} - x, & \bar{q}_{j,k} = g_{j,L} \\ \bar{q}_{j,k} + x, & \bar{q}_{j,k} = g_{j,R} \end{cases} \tag{7}$$

Equation (7) shows that if a bit of value 0 is to be embedded, the value of the cover pixel remains unchanged. However, if a value of 1 is to be embedded, then depending if the value of $\bar{q}_{j,k}$ matches that of $g_{j,L}$ or $g_{j,R}$, the value is modified by ± 1. Otherwise, pixels that do not match $g_{j,L}$ or $g_{j,R}$ are either maintained or shifted by one unit using Eq. (8).

$$q'_{j,k} = \begin{cases} \bar{q}_{j,k}, & g_{j,L} < \bar{q}_{j,k} < g_{j,R} \\ \bar{q}_{j,k} - 1, & \bar{q}_{j,k} < g_{j,L} \\ \bar{q}_{j,k} + 1, & \bar{q}_{j,k} > g_{j,R} \end{cases} \tag{8}$$

In Eq. (8), it can be seen that if $\bar{q}_{j,k}$ is between the peak values, then it remains unchanged, however, if $\bar{q}_{j,k}$ is below $g_{j,L}$, then it is shifted by −1, and by +1 if above $g_{j,R}$. The resulting embedded blocks then make up the final embedded image $I'_e = \{B'^e_j\}_{j=1}^{l}$.

Please note that to make sure the embedded data can be extracted both from the cipher domain and the plaintext domain, not all of the encrypted blocks are applicable to carry data. The smoothness of an encrypted block $B^e_j = \{\hat{q}_{j,L},\ \hat{q}_{j,R},\ \bar{q}_{j,k}\}_{k=1}^{u\times v-2}$ is evaluated by the difference value between the minimal pixel and the maximum pixel. If it is not more than a preset threshold T, as shown in Eq. (9), the block is appropriate to embed data and accordingly a value of '1' is appended to the location map vector H. Otherwise, '0' is appended to H.

$$\max\{\hat{q}_{j,L},\ \hat{q}_{j,R},\ \bar{q}_{j,k}\}_{k=1}^{u\times v-2} - \min\{\hat{q}_{j,L},\ \hat{q}_{j,R},\ \bar{q}_{j,k}\}_{k=1}^{u\times v-2} \le T \tag{9}$$

2.3 Data Extraction and Image Recovery

Given a marked encrypted image $I'_e = \{B'^e_j\}_{j=1}^{l}$ with data embedded as described in the previous section, this section describes the process of extracting embedded data and recovering the image. By this it is meant that the embedded data A can be extracted from the cipher domain I'_e before image decryption and also can be extracted from the decrypted version I' after image decryption. With both of the keys K_c and K_d, the original image I can be reconstructed error-free.

For simplicity, let $B''_j = \{\hat{q}_{j,L},\ \hat{q}_{j,R},\ q''_{j,k}\}_{k=1}^{u\times v-2}$ be the marked blocks with data embedded. Please note that $\{B''_j\}_{j=1}^{l}$ can be the cipher version I'_e and also can be the plaintext version I' decrypted from I'_e, where $K = \{k_j\}_{j=1}^{l}$ is a randomly generated key-stream sized l using RC4 from a secret seed S_c and the content-owner key $K_c = \{u, v, S_c\}$.

$$\begin{aligned} I' &= d(I'_e, K) \\ &= (I'_e - 1 - K)\ mod\ 254 \end{aligned} \tag{10}$$

To extract data from $\{B''_j\}_{j=1}^{l}$, we consider the non-basic pixels $\{q''_{j,k}\}_{k=1}^{u\times v-2}$ in each block. However, it is important to note that we already know the location of basic pixels by the seed S_d from $K_d = \{u, v, S_d, H\}$, and so these pixels are left untouched. The embedded data can be extracted from each block $B''_j = \{\hat{q}_{j,L},\ \hat{q}_{j,R},\ q''_{j,k}\}_{k=1}^{u\times v-2}$ using Eq. (11). Essentially, this means that if the value of non-basic pixel $q''_{j,k}$ is equal to either peak, then it is assumed that data is embedded, and therefore a '0' is extracted. If the value is equal to either $g_{j,L} - 1$ or $g_{j,R} + 1$, then a '1' is extracted.

$$x = \begin{cases} 0, & q''_{j,k} = g_{j,L} \text{ or } q''_{j,j} = g_{j,R} \\ 1, & q''_{j,k} = g_{j,L} - 1 \text{ or } q''_{j,k} = g_{j,R} + 1 \end{cases} \tag{11}$$

In addition to recovering the original signal, the local histogram modification process is also reversed to return the non-basic pixels $q''_{j,k}$ to their unmodified state $\bar{q}_{i,j}$. This is performed as shown in Eq. (12).

$$\bar{q}_{j,k} = \begin{cases} q''_{j,k}, & g_{j,L} < q''_{j,k} < g_{j,R} \\ q''_{j,k} + 1, & q''_{j,k} < g_{j,L} \\ q''_{j,k} - 1, & q''_{j,k} > g_{j,R} \end{cases} \tag{12}$$

3 Experiments and Results

To evaluate RDHEI, there are 4 well known and widely used key indicators: payload, quality of the directly decrypted image, number of incorrectly extracted bits and the reversibility of the original image (error rate). In this section, we conduct a number of different experiments to evaluate the performance of the proposed algorithms. We firstly show the performance of image encryption. Furthermore, the performance of the proposed RDHEI is analyzed and compared with state-of-the-art alternative approaches in terms of the payload, image quality and error rate with several commonly used standard test images.

3.1 Performance of Image Encryption

The histograms corresponding to the associated gray level pixel values before and after encryption are shown in Fig. 2, showing the original image (top row), permutation encryption by [20] (2nd row), stream cipher approach adopted by [15–19] (3rd row), and RC4 adopted in our approach with $u \times v = 2 \times 2$ (bottom row). Since the image encryption schemes introduced in [15–19] are the same, with a stream cipher adopted and applied to all bits of each pixel, the results are the same. It can be seen that, with regard to histogram distribution, leakage of the image histogram is inevitable in Ref. [20], and the image encryption method in this paper has the same uniform appearance as Refs. [15–19].

3.2 Image Quality and Payload Comparison

The payload is the number of bits embedded in each pixel and the unit of measurement is either bits or bpp (bits per pixel). The image quality is often evaluated by the PSNR (peak signal to noise ratio) between the marked and original image. As discussed previously, RDHEI is an emerging technology, but the reported small payload limits its potential for practical applications. Take Lena as an example, the maximum payload of

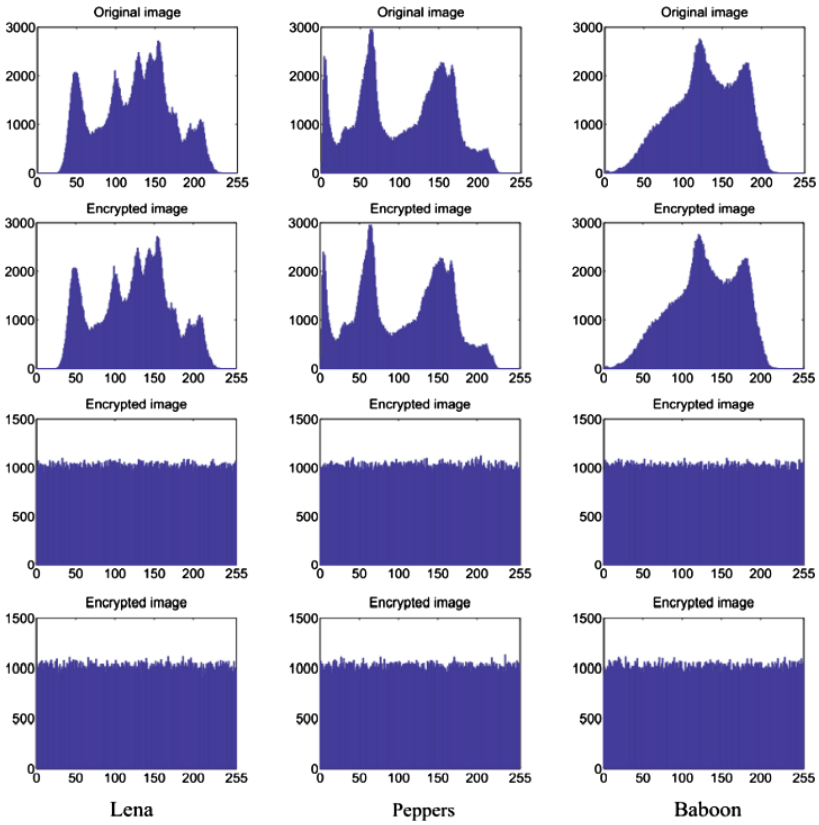

Fig. 2. Gray-level frequency histograms, showing original images (top row), permutation encryption by [20] (2nd row), stream cipher approach by [15–19] (3rd row), and RC4 adopted in our approach (bottom row).

RDHEI proposed by [18] is 8596 bits, about 0.0328 bpp. The payload of the separable method proposed by [19, 20] is much higher, but neither the image quality of [19] nor the security of [20] is satisfactory. In order to prove the value of our proposed method, Fig. 3 shows the PSNR of directly decrypted images generated by Refs. [18, 19] and the proposed method tested on Lena. All results in Fig. 3 are derived from the best parameters under a condition that the embedded data can be extracted exactly and the original image can be recovered error-free. From Fig. 3 we can see that the rate distortion performance of the proposed scheme is the best.

The final indicator, the reversibility of the original image, is the possibility of lossless recovery, and its maximum value is 1 (i.e. fully recovered). If a receiver has both keys, the original image ought to be recovered without error. However, not all images can be fully recovered in Ref [19]. Tables 1 and 2 show the error rate of image recovery in [19] and our proposed method. To get the best results, the 8-th bit of the host pixel is used to embed data in Wu's method. And we perform the experiment

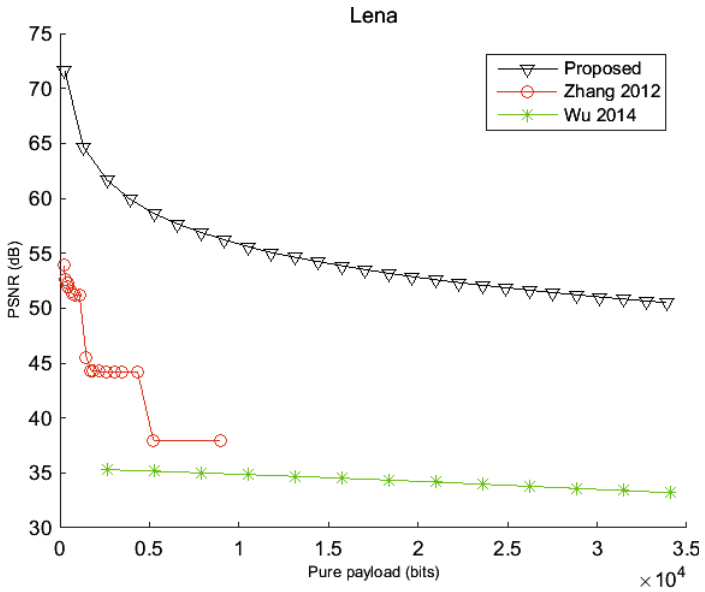

Fig. 3. Image quality and payload comparison for Lena, showing the PSNR of directly decrypted images generated by Refs. [18, 19], and the proposed method.

Table 1. Reversibility comparison on Sailboat ($T = 188$).

Methods	Payload (bpp)	PSNR (I, I')	Error rate
Proposed	0.01	58.6	0
[19]		30.63	0.1
Proposed	0.03	53.89	0
[19]		30.46	0.3
Proposed	0.04	51.29	0
[19]		30.38	0.36

Table 2. Reversibility comparison on Jet ($T = 8$).

Methods	Payload (bpp)	PSNR (I, I')	Error rate
Proposed	0.04	56.46	0
[19]		33.95	0.06
Proposed	0.08	56.07	0
[19]		33.15	0.1
Proposed	0.12	55.49	0
[19]		32.27	0.18

in each image 100 times with key from 1 to 100 to calculate the mean error rate. All experimental results show that the error rate of image in the proposed method is always 0, better than Ref. [19].

4 Conclusion

This paper proposed and evaluated a complete separable framework for reversible data hiding in encrypted images. The embedded data can be extracted error-free both from the cipher domain and the plaintext domain. However, the proposed method is not suitable for images containing saturated pixels. Future work will aim to improve this.

Acknowledgements. This research work is supported by National Natural Science Foundation of China under Grant Nos. 61502009 and 61472235, Anhui Provincial Natural Science Foundation under Grant No. 1508085SQF216, the 48th Scientific Research Staring Foundation for the Returned Overseas Chinese Scholars, Ministry of Education of China under Grant No. 1685 and the Foundation of Center of Information Support and Assurance Technology for Anhui University under Grant No. ADXXBZ201411. The authors appreciate Dr. Andrew Abel from the University of Stirling for proofreading.

References

1. Hong, W., Chen, T.S.: A novel data embedding method using adaptive pixel pair matching. IEEE Trans. Inf. Forensics Secur. **7**(1), 176–184 (2012). doi:10.1109/tifs.2011.2155062
2. Tian, H., Liu, J., Li, S.: Improving security of quantization-index-modulation steganography in low bit-rate speech streams. Multimedia Syst. **20**(2), 143–154 (2014). doi:10.1007/s00530-013-0302-8
3. Celik, M.U., Sharma, G., Tekalp, A.M., Saber, E.: Lossless generalized–LSB data embedding. IEEE Trans. Image Process. **14**(2), 253–256 (2005). doi:10.1109/TIP.2004.840686
4. Peng, F., Li, X., Yang, B.: Adaptive reversible data hiding scheme based on integer transform. Sig. Process. **92**(1), 54–62 (2012). doi:10.1016/j.sigpro.2011.06.006
5. Tian, J.: Reversible data embedding using a difference expansion. IEEE Trans. Circ. Syst. Video Technol. **13**(8), 890–896 (2003). doi:10.1109/tcsvt.2003.815962
6. Ni, Z., Shi, Y.Q., Ansari, N., Su, W.: Reversible data hiding. IEEE Trans. Circ. Syst. Video Technol. **16**(3), 354–362 (2006). doi:10.1109/tcsvt.2006.869964
7. Tai, W.L., Yeh, C.M., Chang, C.C.: Reversible data hiding based on histogram modification of pixel differences. IEEE Trans. Circ. Syst. Video Technol. **19**(6), 906–910 (2009). doi:10.1109/tcsvt.2009.2017409
8. Tsai, P., Hu, Y.C., Yeh, H.L.: Reversible image hiding scheme using predictive coding and histogram shifting. Sig. Process. **89**(6), 1129–1143 (2009). doi:10.1016/j.sigpro.2008.12.017
9. Zhang, X., Wang, S., Qian, Z., Feng, G.: Reversible fragile watermarking for locating tempered blocks in JPEG images. Sig. Process. **90**(12), 3026–3036 (2010). doi:10.1016/j.sigpro.2010.04.027
10. Qian, Z., Zhang, X.: Lossless data hiding in JPEG bitstream. J. Syst. Softw. **85**(2), 309–313 (2012). doi:10.1016/j.jss.2011.08.015
11. Erkin, Z., Piva, A., Katzenbeisser, S., Lagendijk, R.L., Shokrollahi, J., Neven, G., Barni, M.: Protection and retrieval of encrypted multimedia content: when cryptography meets signal processing. EURASIP J. Inf. Secur. **2007**, 1–20 (2007). doi:10.1155/2007/78943
12. Schmitz, R., Li, S., Grecos, C., Zhang, X.: Towards robust invariant commutative watermarking-encryption based on image histograms. Int. J. Multimedia Data Eng. Manage. **5**(4), 36–52 (2014). doi:10.4018/ijmdem.2014100103

13. Ma, K., Zhang, W., Zhao, X., Yu, N., Li, F.: Reversible data hiding in encrypted images by reserving room before encryption. IEEE Trans. Inf. Forensics Secur. **8**(3), 553–562 (2013). doi:10.1109/tifs.2013.2248725
14. Zhang, W., Ma, K., Yu, N.: Reversibility improved data hiding in encrypted images. Sig. Process. **94**, 118–127 (2014). doi:10.1016/j.sigpro.2013.06.023
15. Zhang, X.: Reversible data hiding in encrypted image. IEEE Sig. Process. Lett. **18**(4), 255–258 (2011). doi:10.1109/lsp.2011.2114651
16. Hong, W., Chen, T.S., Wu, H.Y.: An improved reversible data hiding in encrypted images using side match. IEEE Sig. Process. Lett. **19**(4), 199–202 (2012). doi:10.1109/lsp.2012.2187334
17. Liao, X., Shu, C.: Reversible data hiding in encrypted images based on absolute mean difference of multiple neighboring pixels. J. Vis. Commun. Image Represent. **28**, 21–27 (2015). doi:10.1016/j.jvcir.2014.12.007
18. Zhang, X.: Separable reversible data hiding in encrypted image. IEEE Trans. Inf. Forensics Secur. **7**(2), 826–832 (2012). doi:10.1109/tifs.2011.2176120
19. Wu, X., Sun, W.: High-capacity reversible data hiding in encrypted images by prediction error. Sig. Process. **104**, 387–400 (2014). doi:10.1016/j.sigpro.2014.04.032
20. Yin, Z., Luo, B., Hong, W.: Separable and error-free reversible data hiding in encrypted image with high payload. Sci. World J. **2014**, 1–8 (2014). doi:10.1155/2014/604876
21. Ferguson, N., Schneier, B.: Practical Cryptography. Wiley, New York (2003)