J. Vis. Commun. Image R. 25 (2014) 322–328

Contents lists available at ScienceDirect

J. Vis. Commun. Image R.

journal homepage: www.elsevier.com/locate/jvci

Efficient reversible data hiding in encrypted images

Xinpeng Zhang *, Zhenxing Qian, Guorui Feng, Yanli Ren

School of Communication and Information Engineering, Shanghai University, 149 Yanchang Road, Shanghai 200072, PR China

ARTICLE INFO

Article history:
Received 29 June 2013
Accepted 2 November 2013
Available online 9 November 2013

Keywords:
Reversible data hiding
Image encryption
Embedding payload
LDPC code
Side information
Data compression
Least significant bits (LSB)
Most significant bits (MSB)

ABSTRACT

This paper proposes a novel scheme of reversible data hiding in encrypted images based on lossless compression of encrypted data. In encryption phase, a stream cipher is used to mask the original content. Then, a data hider compresses a part of encrypted data in the cipher-text image using LDPC code, and inserts the compressed data as well as the additional data into the part of encrypted data itself using efficient embedding method. Since the majority of encrypted data are kept unchanged, the quality of directly decrypted image is satisfactory. A receiver with the data-hiding key can successfully extract the additional data and the compressed data. By exploiting the compressed data and the side information provided by the unchanged data, the receiver can further recover the original plaintext image without any error. Experimental result shows that the proposed scheme significantly outperforms the previous approaches.

1. Introduction

Data hiding is a technique to embed additional information into digital multimedia by slightly altering the cover signals. When the data hiding is performed with a reversible manner, the original cover content can be perfectly restored after data extraction at receiver side. The reversible data hiding methods may be roughly classified into three types: the difference expansion methods, histogram modification methods, and lossless compression based methods. In the difference expansion methods [1–3], the differences between two adjacent pixels are doubled to generate a new least significant bit (LSB) plane for embedding the additional information. The histogram modification methods shift the histogram of cover data from its peak point towards its zero points, and utilize the cover data at the peak point of histogram to carry the additional data [4,5]. As the third type of reversible data hiding, the lossless compression based methods make use of statistical redundancy of the host media by performing lossless compression so that a spare space for accommodating the additional data can be created [6].

Most of works on reversible data hiding are suitable for unencrypted covers. Recently, the reversible data hiding for encrypted images has been also studied. In some application scenarios, a content owner aims to encrypt original images as unintelligible data for privacy protection. For an inferior assistant or a channel administrator who does not know the cryptographic key and the plaintext content, he may hope to append some additional data, such as the origin information, image notation or authentication data, within the cipher-text images. Clearly, the plaintext content and the appended data should not be accessible to unauthorized parties. On the other hand, for a legal receiver, it is also desired that the original plaintext content can be recovered without any error after image decryption and data extraction. That implies there are three participants in the scenarios: content owner, data hider and legal receiver. In [7], each block of cipher-text image encrypted by AES algorithm is used to carry one additional bit with a simple substitute method, and an analysis of local standard deviation of the marked encrypted images is used to remove the embedded data during the decryption step. However, the payload is low and the image decrypted from the marked encrypted version is seriously degraded. In [8], the data hider modifies only the LSB of cipher-text image encrypted by exclusive-OR operation, so that the quality of directly decrypted image is good. With the aid of spatial correlation in natural image, the embedded data can be successfully extracted and the original image can be perfectly recovered from the directly decrypted image. The performance is further improved by introducing a block-recovery order sorted by the smoothness and a side match mechanism [9]. In [8,9], the additional data must be extracted from the decrypted image, in other words, if a receiver has the data-hiding key but not the cryptographic key, he cannot extract any information from the encrypted image containing additional data. Using the separable scheme presented in [10], with an encrypted image containing additional data, a receiver having the data-hiding key can extract the additional data, while a receiver having the cryptographic key can decrypt

* Corresponding author. Fax: +86 21 56331435.
E-mail address: xzhang@shu.edu.cn (X. Zhang).

1047-3203/$ - see front matter © 2013 Elsevier Inc. All rights reserved.
http://dx.doi.org/10.1016/j.jvcir.2013.11.001

the received data to obtain an image similar to the original one. If the receiver has both the data-hiding and cryptographic keys, he can extract the additional data and recover the original image without any error.

As a means used in reversible data hiding for unencrypted covers, lossless compression is potentially applicable to reversible data hiding for encrypted images. Compression of encrypted data has been investigated in [11], and it is also proved that the performance of compressing encrypted data may be as good as that of compressing non-encrypted data in theory [11]. With the practical method presented in [11], the encrypted binary images can be losslessly compressed by finding the syndromes of low-density parity-check (LDPC) codes [12]. In another lossless compression method [13], the encrypted gray image is decomposed in a progressive manner, and the data in most significant bit (MSB) planes are compressed using rate-compatible punctured turbo codes. At received side, the original content can be perfectly reconstructed based on the statistics of local content. By exploiting the statistical models of digital video, some algorithms for compressing encrypted video are also developed [14]. In addition, a quantization-based lossy compression method for encrypted images and a corresponding iterative image reconstruction method are proposed in [15].

This paper proposes a novel scheme of reversible data hiding in encrypted images, in which a part of encrypted data is losslessly compressed using LDPC code and used to carry the compressed data as well as the additional data. Since the majority of encrypted data are kept unchanged, the quality of directly decrypted image is satisfactory. Furthermore, the side information provided by the unchanged data can be used to perfectly recover the original content after image decryption and data extraction. Experimental result shows that the proposed scheme significantly outperforms the previous approaches.

2. Proposed scheme

In the proposed scheme, the content owner encrypts the original image using a stream cipher. Although the data hider does not know the original content and the cryptographic key, he may compress a half of the 4th LSB in the encrypted image using LDPC code, and then insert the compressed data together with the additional data into the encrypted image using an efficient embedding method. When getting an encrypted image containing embedded data, the receiver can extract the additional data using the data-hiding key, or decrypt it using the cryptographic key to obtain an image similar to the original version. If the receiver has both the data-hiding and cryptographic keys, he can further recover the original image without any error.

2.1. System description

Assume the original plaintext image is in uncompressed format and each pixel is represented by 8 bits. The content owner may use a stream cipher to generate an encrypted version,

$$\mathbf{I}_{\mathrm{E}} = \mathbf{I}_{\mathrm{O}} \oplus \mathbf{K}_{\mathrm{E}} \tag{1}$$

where $\mathbf{I}_{\mathrm{O}}$, $\mathbf{K}_{\mathrm{E}}$ and $\mathbf{I}_{\mathrm{E}}$ are the data of original image, the key stream and the encrypted data respectively, and '$\oplus$' denotes the bit-wise exclusive OR operation. That implies, in the encryption phase, the original data are masked and their positions are not changed.

With the encrypted image $\mathbf{I}_{\mathrm{E}}$, although the data hider does not know the original content and the key stream, he may insert some additional data into $\mathbf{I}_{\mathrm{E}}$ in a reversible manner by employing the encrypted-data compression and efficient data-embedding mechanisms. As shown in [13], the 3 LSB of encrypted images are hard to be compressed due to the lack of statistical redundancy in plaintext content. On the other hand, in order to ensure that the image quality after decryption is not seriously affected by the data-embedding operations, the encrypted data in high bit-planes should be not changed. Considering the two aspects, we choose a half of fourth LSB as the space to carry the data to be embedded. In the data embedding procedure, the pixels in encrypted image are firstly divided into two sets in a chessboard fashion as shown in Fig. 1, and we denote the fourth LSB at the white positions as $\mathbf{I}_E^{(4W)}$. While $\mathbf{I}_E^{(4W)}$ is used to carry the additional data and a compressed version of $\mathbf{I}_E^{(4W)}$ itself, the rest data in the encrypted image, which include all the four most signification bits (MSB), all the three LSB and the fourth LSB at black positions, are kept unchanged. Actually, the unchanged data will be exploited to provide side information for the recovery of original content at receiver side.

Fig. 2 gives a sketch of the data embedding procedure. For compressing $\mathbf{I}_E^{(4W)}$, the data hider permutes and reorganizes $\mathbf{I}_E^{(4W)}$, which is a part of encrypted data, as K blocks with length n_c, $\mathbf{v}_k = [v_k(1), v_k(2), \ldots, v_k(n_c)]^{\mathrm{T}}$ $(1 \leqslant k \leqslant K)$, and calculates

$$\mathbf{u}_k = \mathbf{H} \cdot \mathbf{v}_k, \quad k = 1, 2, \ldots, K \tag{2}$$

where $\mathbf{H}$ is an $n_{\mathrm{r}} \times n_{\mathrm{c}}$ check matrix of a low-density parity-check (LDPC) code and the arithmetic is modulo-2 ($n_{\mathrm{r}} < n_{\mathrm{c}}$). The matrix $\mathbf{H}$ can be generated from some LDPC code designing methods, such as [16–18], and the generation way is shared by the data hider and the receiver. This way, the vectors $\mathbf{u}_k$ are regarded as the compressed data of $\mathbf{v}_k$, and the compression ratio is

$$r = n_{\mathrm{r}}/n_{\mathrm{c}} \tag{3}$$

Denoting the pixel number of cover image as N, the bit amount of all $\mathbf{u}_k$ is $r{\cdot}N/2$. The payload to be embedded into $\mathbf{I}_E^{(4W)}$ is made up of all $\mathbf{u}_k$, the values of n_{r} and n_{c}, and the additional data. In the following, we ignore the data amount for representing n_r and n_c since it is very small. For embedding the payload, a proportion of bits in $\mathbf{I}_E^{(4W)}$ are inevitable to be flipped. Denote the ratio between the bit amounts of the payload and $\mathbf{I}_E^{(4W)}$ as α, and the ratio between the number of flipped bits and the bit amount of $\mathbf{I}_E^{(4W)}$ as β. In data hiding applications, it is often desired to maximize the embedded payload with a given distortion level. As pointed out by [19], there must have α is not more than the binary entropy of β,

$$\alpha \leqslant H(\beta) = -\beta \cdot \log_2\beta - (1-\beta) \cdot \log_2(1-\beta) \tag{4}$$

A number of practical data embedding methods approaching the payload bound have been developed [19–21], in which α may be very close to $H(\beta)$. Thus, the data hider may employ a practical method with the parameters (α, β) to embed the payload into $\mathbf{I}_E^{(4W)}$. Here, a data-hiding key can be also employed to determine a walking way in the bits of $\mathbf{I}_E^{(4W)}$ to ensure any attacker without the data-hiding key cannot extract the embedded data. Clearly, a necessary condition of successfully embedding the additional data is that $\alpha > r$, and, in this case, the pure embedding rate, a ratio between the bit amount of the additional data and the pixel number of cover image, is

$$R_P = (\alpha - r)/2 \tag{5}$$

The efficient data-embedding method with $\alpha \approx H(\beta)$ used here may lead to a high R_{P}. We denote the new version of $\mathbf{I}_E^{(4W)}$ as $\mathbf{I}_E^{\prime(4W)}$. By

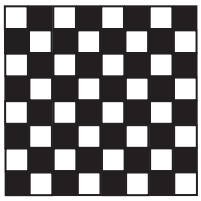

Fig. 1. Pixel division in chessboard fashion.