Contents lists available at ScienceDirect

Signal Processing

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

Binary-block embedding for reversible data hiding in encrypted images

Shuang Yi, Yicong Zhou*

Department of Computer and Information Science, University of Macau, Macau 999078, China

ARTICLE INFO

Keywords:
Binary-block embedding (BBE)
Reversible data hiding
Image encryption
Encrypted domain

ABSTRACT

This paper first introduces a binary-block embedding (BBE) method to embed secret data in a binary image. Using BBE, we propose an algorithm for reversible data hiding in encrypted images (BBE-RDHEI). It uses BBE to embed binary bits in lower bit-planes of the original image into its higher bit-planes such that the lower bit-planes can be reserved for hiding secret data in subsequent processes. BBE-RDHEI employs a bit-level scrambling process after secret data embedding to spread embedded secret data to the entire marked encrypted image so that it can prevent secret data from loss. A security key design mechanism is proposed to enhance the security level of BBE-RDHEI. The processes of BBE-RDHEI are fully reversible. The secret data and original image can be reconstructed independently and separately. Experiments and comparisons show that BBE-RDHEI has an embedding rate nearly twice larger than the state-of-the-art algorithms, generates the marked decrypted images with high quality, and is able to withstand the brute-force, differential, noise and data loss attacks.

1. Introduction

Reversible data hiding (RDH) is a technique that slightly alters digital media (e.g. images or videos) to embed secret data while the original digital media can be completely recovered without any error after the hidden messages have been extracted [1]. It is quite useful for various applications in military, medical science or law enforcement, where the original images or videos should not be damaged. A number of RDH methods were proposed in recent years. Histogram shifting (HS) shifts several or the maximum points in histogram bins of the original image to reserve spare space for data embedding [1]. To improve the embedding capacity, prediction-error based HS algorithms were introduced [2–4]. Difference expansion (DE) [5–7] as another type of RDH divides the image into pixel pairs and embeds secret data into the expanded difference values. Integer transforms have been used to modify the values of pixel pairs to embed secret data [8–10]. These RDH methods need the redundancy information of image pixels in original images to embed secret data, such as the statistic or difference information of pixel pairs. They are not suitable for encrypted images that are noise-like and have no redundancy information available.

Recently, reversible data hiding in encrypted images (RDHEI) has attracted people's attention. It aims to protect both the original images and secret data simultaneously. For example, the content owner intends to store an original image in the Cloud that is hosted by a third party. To prevent the content of the original image from being exposed to the third party, the content owner encrypts the image before sending it to the Cloud. Meanwhile, the system administrator of the third party is able to add some notations to the encrypted image without knowing its original content. Depending on whether the data extraction and image recovery processes can be performed separately, existing RDHEI methods can be classified into joint and separate methods.

For joint methods, Peuch et al. [11] first encrypted each block of the original image by the advanced encryption standard (AES) and then embedded one bit of the secret data into each encrypted block by bit substitution. The encrypted image embedded with secret data is called the marked encrypted image. Secret data extraction is just to obtain the bits in the substituted positions. Original image recovering is accomplished by analyzing the local standard deviation of the marked encrypted image during the decryption procedures. This algorithm has a limited payload and yields the decrypted image with low quality. Another joint RDHEI algorithm proposed by Zhang [12] encrypts the original image using bit-level XOR, and then embeds one bit of secret data into each block of the encrypted image by shifting the three least significant bits (LSBs) of half pixels within the block. This algorithm may suffer from incorrect results of data extraction and image recovering in the non-smoothness regions in the image when the block size is relatively small (e.g., 8×8). Hong et al. [13] proposed an improved version of this algorithm by modifying its smoothness measurement function. The error rate of data extraction is reduced for small block sizes. In Wu et al.'s joint method [14], one bit of the secret data is embedded by flipping the i^{th} $(1 \le i \le 6)$ bit of pixels in a

* Corresponding author.
E-mail address: yicongzhou@umac.mo (Y. Zhou).

http://dx.doi.org/10.1016/j.sigpro.2016.10.017
Received 13 May 2016; Received in revised form 6 September 2016; Accepted 12 October 2016
Available online 25 October 2016

certain group. This method also may suffer from incorrect results of data extraction and image recovery.

To allow the receiver with different privileges to obtain different contents (the secret data, the original image or both) from the marked encrypted image, researchers devote themselves to develop separable RDHEI methods. Zhang et al. [15,16] proposed two separable methods that compress the encrypted image to accommodate secret data. In Wu et al.'s separable method [14], one bit of the secret data is embedded by replacing the i^{th} ($i \geq 7$ for the later one) bit of pixels in a certain group. Secret data extraction and image recovering are using the prediction error. Compared with algorithms in [12,13,15], the methods in [14] reduce the number of incorrectly extracted secret data bits and improve the visual quality of the marked decrypted image. Qian [17] proposed a separable RDHEI algorithm using n-nary histogram modification. However, it results in Salt & Pepper noise in the marked encrypted images. Besides, instead of working on the spatial domain, Qian et al. [18] proposed an RDHEI method to embed secret data in the encrypted JPEG bitstream. In [19] and [20], homomorphic encryption is utilized to encrypt the original image. However, image size increases because the used homomorphic encryption algorithm maps the pixel value into a larger data range. In above mentioned RDHEI methods, the content owner does nothing except for image encryption. These methods have a small payload and/or a high error rate in data extraction and image recovering.

To overcome these problems, some researchers aim to develop another type of separable RDHEI method by reserving the spare space for secret data embedding before image encryption. In Ma's method [21], it reserves the spare space by embedding some LSBs in a part of the cover image into the rest part of the cover image with using simplified RDH method in [22]. The self-embedding of LSBs ensures the reversibility of image recovering. Zhang et al. [23] selected some pixels and applied a histogram shifting method to their estimation error values for accommodating secret data.

Previous RDHEI methods in [12,13,15,14,21] have a limited embedding rate, and are under the only situation that the images are for the Cloud storage with no transmission involved and thus no attacks [18]. Considering the scenario that hospitals at different locations build a bridge for cooperations, many medical images embedded with patients' information or treatment history records will be shared among several working teams, thus medical images will be transmitted over public channels that they may inevitably experience some noise and data loss. In this scenario, these RDHEI methods may suffer from secret data loss when the marked encrypted image is partially damaged or lost. For example, method in [21] uses a part of the LSB plane in the encrypted image to accommodate the secret data when embedding rate is less than 0.2 bpp. If the LSB plane is illegally removed, all secret data will lose. In the separable method in [14], the secret data are embedded in the i^{th} most significant bit (MSB) plane, it will also suffer from complete loss of secret data when this MSB plane is removed or damaged.

To improve the embedding rate while enhancing security and robustness, this paper introduces the binary-block embedding (BBE) method to embed message bits in binary images. Based on BBE, we further propose a reversible data hiding algorithm in encrypted images (BBE-RDHEI). It first uses BBE to embed binary bits in several LSB planes of the original image into its MSB planes. BBE-RDHEI encrypts the original image and hides the secret data into its LSB planes. A bit-level scrambling process is then employed after secret data embedding to ensure that the proposed BBE-RDHEI can resist the noise and data loss attacks. Using different security keys, the receiver is able to obtain the secret data, marked decrypted image, decrypted image, or all of them from the marked encrypted image.

Our main contributions in this work are listed as follows:

(1) We propose a new BBE algorithm for reversible data hiding in the encryption domain, which is totally different from traditional RDH methods. BBE can be utilized in different types of images such as binary, gray-scale, medical and cartoon images.

(2) Based on BBE, we further propose a method of reversible data hiding in encrypted images, BBE-RDHEI. Compared with existing state-of-the-art methods, it has significantly improved embedding capacity and quality of the marked decrypted image. BBE-RDHEI can also be simplified and utilized for binary images, while existing RDHEI methods are designed only for gray-scale images.

(3) To significantly enhance the security level of BBE-RDHEI, we also propose a security key design mechanism such that BBE-RDHEI is able to resist the differential attack, while existing RDHEI methods cannot.

(4) To enhance the robustness of RDHEI methods in withstanding noise and data loss attacks, we introduce a bit-level scrambling process to BBE-RDHEI after secret data embedding to spread out embedded secret data over the entire marked encrypted image. As a result, BBE-RDHEI is able to recover most of secret data even if one bit-plane (e.g., LSB or MSB) of the marked encrypted image is completely removed. Moreover, any bit-level scrambling algorithm can be used in our BBE-RDHEI. This is another security benefit of BBE-RDHEI.

The rest of this paper is organized as follows: Section 2 will introduce the BBE algorithm. Section 3 will propose BBE-RDHEI. Simulation results and comparisons will be provided in Section 5. Section 6 will provide security and robustness analysis of the proposed BBE-RDHEI. Section 7 will draw a conclusion.

2. Binary-block embedding

In this section, we propose a binary-block embedding (BBE) algorithm to embed message bits into a binary image.

2.1. BBE

BBE first divides the binary image into a number of non-overlapping blocks, separates them into two groups named good and bad blocks, respectively. A good block is able to be embedded with messages while a bad one is not. In message embedding phase, BBE first labels the first 2 or 3 bits of each block with special bits that indicate the block types. Then the rest bits of a good block will be replaced with its structure information and message bits while the rest bits of a bad block will be kept unchanged. Next, we present the BBE algorithm in detail.

2.1.1. *Block labeling*

Assume that a binary image with a size of $M \times N$ is able to embed secret data. We first divide the image into a set of non-overlapping blocks with a size of $s_1 \times s_2$, where $s_1, s_2 \geq 3$. For a certain block, we let $n = s_1 * s_2$ be the total number of pixels within the block, and $m = \min\{n_0, n_1\}$ be the minimum value of n_0 and n_1, where n_0 and n_1 are the numbers of 0 s and 1 s within the block, respectively. According to a threshold n_α, we then classify these blocks into five categories as shown in Table 1, namely: Good-I/II/III/IV block and Bad block, where a good block is able embed secret data, while a bad one cannot.

Table 1
Block types and block-labeling bits.

Condition	Block type	Block description	Block-labeling bits
$m > n_a$	Bad	cannot embed data	00
$m = n_0 = 0$	Good-I	all pixels are 1	11
$m = n_1 = 0$	Good-II	all pixels are 0	10
$1 \leq m \leq n_a, n_0 < n_1$	Good-III	most of pixels are 1	011
$1 \leq m \leq n_a, n_1 < n_0$	Good-IV	most of pixels are 0	010