A New Steganography Scheme based on an Index-color Image

Se-Min Kim
Multimedia Track,
School of Engineering,
Information and
Communications Univ.,
Daejeon, South Korea
resemin@icu.ac.kr

Ziqiang Cheng
Graduate School of EECS,
Kyungpook National Univ.,
Daegu, South Korea
ziqiangcheng@infosec.knu.ac.kr

Kee-Young Yoo *
Dept. of Computer Eng.,
Kyungpook National Univ.,
Daegu, South Korea
yook@knu.ac.kr

Abstract

In this paper, we propose a new more efficient steganography technique that can be applied in Internet environment. The new efficient technique can hide from 1 to 8 bits secret data per pixel, and has no distortion if the number of colors does not exceed 128. The proposed scheme first divides the secret data into several parts based on the number of colors in the cover image, and then embed secret data into the cover image, part by part by expanding palettes and modifying indices. Finally, the cover image can be recovered easily without loss. Indeed, numerical experiments indicate that the new technique introduces no distortion to the cover image, in contrast with other schemes that based on index-color, such as EZ Stego. Furthermore, the proposed scheme can be applied to BMP, GIF and PNG image formats which use the index-color technique. Thus, the proposed scheme suits the current Internet environment very well.

Key Words: steganography, index-color, palette, distortion, capacity

1 Introduction

Steganography is the art and science of hiding communication, a steganographic system thus embeds hidden content in unremarkable cover media so as not to arouse an eavesdropper’s suspicion [6]. A long time ago, the most popular steganographic methods used by spies include invisible ink and microdots. Today, it seems natural to use digital images, digital video, or audio for hiding secret data [2]. In the case of images, the carrier is referred to as the cover image, while after embedding secret data into it, the stego image can be obtained. Additionally, to make the secretly hidden information difficult to suspect and detect by any eavesdroppers while the stego image is transmitted through the unsecure channel, the secret data can be compressed and encrypted before it is hidden in the carrier. This is important because in this way we minimize the amount of information that is to be sent, and it is also easier to hide a random looking message into the carrier than to hide a message with a high degree of regularity.

Generally, we have to consider two principles while performing steganography. The first one is the capacity of hidden secret data, and another is the quality of the stego image. Many techniques for steganography have tried to catch both. But this is not easy because the two principles are antithetic to each other. If we hide a lot of secret data in a cover image, the quality of the stego image becomes low. On the contrary, if we hide little secret data in a cover image, the quality of the stego image becomes high. Thus, we should balance the above two principles and try to find an optimum method.

The common well-known steganographic method is the least-significant-bits (LSBs) substitution which was proposed by Bender et al. [7]. Wang et al.’s scheme [8] is also based on LSB but provides better image quality for stego image. These methods replace the fixed-length LSBs of pixels with the embedding data. However, high distortion probably occurs. One of the most popular message hiding schemes for index-color images has been proposed by Machado, called EZ Stego [5]. Fridrich [3] presented another new steganographic technique for hiding secret data in index-color image based on the parity bits. Although Fridrich’s scheme introduces far less severe distortion than EZ Stego, the two schemes only embed 1 bit per pixel, in other words, they provide a low embedding capacity.

In this paper, the goal of the proposed steganographic technique is to provide high embedding capacity and low distortion. Based on the number of colors (NOC) in a cover

*Corresponding author: Kee-Young Yoo (Tel.: +82-53-950-5553; Fax: +82-53-957-4846)
2. Preliminary

In this section, we introduce the theory of index-color and review EZ Stego, Fridrich’s schemes in brief.

2.1 Index-color

There are several representative image file formats that support index-color modes, such as GIF, BMP and PNG. Although these formats have different file types and compression methods, they have a similar structure. Table 1 shows the general structure of an index-color image [4]. Basic properties (such as image type, image size, palette offset) of an image are stored in ‘Image Header’. The colors which compose the image are in ‘Palette’. Table 2 shows the common palette entry structure of a gray-level image. Each palette entry consists of an index and a color. The number of palette items is related to the \( \text{NOC} \) of an image, for instance, 256 colors (8-bit) BMP image holds 256 palette entries. In the ‘Image Data’ part, we can find all indices of image.

<table>
<thead>
<tr>
<th>Index</th>
<th>Color</th>
</tr>
</thead>
<tbody>
<tr>
<td>00</td>
<td>(00, 00, 00)</td>
</tr>
<tr>
<td>01</td>
<td>(01, 01, 01)</td>
</tr>
<tr>
<td>:</td>
<td>:</td>
</tr>
<tr>
<td>FE</td>
<td>(FE, FE, FE)</td>
</tr>
<tr>
<td>FF</td>
<td>(FF, FF, FF)</td>
</tr>
</tbody>
</table>

2.2 EZ Stego scheme

One of the most popular message hiding schemes for index-color based image has been proposed by Machado, called EZ Stego [5], in which secret data can be embedded by modifying the index value of the cover image. It does not directly embed secret data into LSB of indices since that maybe introduces a high distortion. Therefore, the palette is first sorted by luminance using equation (1), where \( R \), \( G \), \( B \) represent Red, Green, Blue color value, respectively.

\[
Luminance = (0.299R + 0.587G + 0.144B) \quad (1)
\]

In the reordered palette, neighboring palette entries are typically near to each other in the color spaces, as well. Then EZ Stego embeds the secret data in a binary form into the LSB of the indices pointing to the palette colors. EZ Stego is based on the premise that close colors in the luminance-ordered palette are close in the color space. But actually colors with similar luminance values may be represent two completely different colors, which might lead to high distortion of the stego image.

2.3 Fridrich’s scheme

Fridrich [3] proposed another steganographic scheme on the basis of the parity of close colors in order to solve the deterioration problem of EZ Stego. In Fridrich’s scheme, the Euclidean norm distance between the colors of the pixel from each palette entry is first calculated by using equation (2). The Euclid distance \( d \) between colors \( (R_1, G_1, B_1) \) and \( (R_2, G_2, B_2) \) is,

\[
d = \sqrt{(R_1 - R_2)^2 + (G_1 - G_2)^2 + (B_1 - B_2)^2} \quad (2)
\]

Each color corresponds to a parity of the color, the closest colors are searched in the palette until a match is found between the bit to be embedded and the parity of the color, where the parity of a color is defined as \( (R + G + B) \mod 2 \). Once the color is found, the index for the pixel is changed to point to the new color. Then the secret data can be simply read by extracting the parity bits of the colors of selected pixels.

Even though Fridrich’s scheme obviously avoid the discontinuities that occasionally happen in EZ Stego, it only embeds one secret data bit per pixel which is the same as EZ Stego scheme, and provides a low embedding capacity therefore.

3 Proposed Method

In this section, we propose a new steganographic technique for index-color images, such as GIF, BMP, PNG and so on.
3.1 Embedding

The embedding process is shown as (a) in Fig. 1.

![Flowcharts of proposed scheme: (a) Embedding process (b) Extracting process](image)

3.1.1 Calculating $SBL$

The proposed scheme is based on the theory that index-color mode supports from 1-bit to 8-bit for each pixel, hence the current index can be expanded to 8-bit, for embedding secret data. First of all, the $SBL$ has to be calculated according to the $NOC$ of the cover image. Where $NOC$ refers to the number of colors, and $SBL$ means the length of the secret data which will be embedded into each pixel. $SBL$ can be calculated by using equation (3). For example, if the $NOC$ of the cover image is 4, the corresponding $SBL$ is 6, meaning that 6-bit secret data can be embedded into each pixel.

$$SBL = 8 - \lceil \log_2 NOC \rceil$$  \hspace{1cm} (3)

The proposed scheme is invalid when the $NOC$ exceeds 128, since there are no spaces for embedding secret data in each pixel index. However, it still can be applied to a gray-level or gradation image, when the $NOC$ of the cover image is reduced to 128, by eliminating odd indices in the cover image.

3.1.2 Expanding palette

In the proposed scheme, embedding secret data into index of the cover image will introduce new index values, so the current palette needs to be expanded for matching new additional indices. The number of palette entries will be 256 after expanding. Fig. 2 shows the expanding operation of a cover image. For simplicity, assuming that there are 4 colors in the cover image, hence palette entries from 04\textsuperscript{th} to FF\textsuperscript{th} are expanded. Fig. 2 (a) is the original palette of the cover image, and (b) represents the expanded palette.

3.1.3 Constructing stego image

First, the secret data is transformed to binary form. It is recommended that encrypt it using AES [1], in order to provide more security. At each pixel position, the right-most $\lceil \log_2 NOC \rceil$ bits of the original index are concatenated with $SBL$ bits of the secret data. After performing concatenating operation, the new index becomes 8 bits, and the secret data can be embedded simply. For example, as shown in Fig. 3, assuming the $NOC$ of the cover image is 4, say $SBL$ is 6. The embedding process starts from the top left corner of the cover image. Suppose the original index value at coordinate (0, 0) is 00000000, the right-most $\lceil \log_2 4 \rceil$ bits are first truncated, and then 6 bits of the secret data are selected in binary form. After concatenating them together, say 00||010000, the new index value 00010000 can be generated. Finally, the new index value is assigned to the cor-
The experiment results for the proposed scheme are illustrated in this section, including the comparison of the proposed scheme with the previous index-color based schemes.

4.1 Experiment results

To demonstrate the result of the proposed scheme, a 256×256 gray-level secret image, see Fig. 5 (c), is cryptographically encoded into 1 color and 4 colors 256 8-bit cover images, see Fig. 5 (a) and (b), respectively.
4.1.1 1 color cover image

The proposed scheme has the best embedding capacity when the cover image has 1 color because 8-bit secret data can be embedded into each pixel of cover image. Fig. 6 (a) is the stego image, the 256\times256 secret image can be hidden in 1 color 8-bit cover image completely. No distortion occurs after embedding, however we gain an increased stego image in capacity since we have expanded the palette entries and modified the indices of the cover image.

4.1.2 4 colors cover image

If cover image has 4 colors, 6-bit secret data can be embedded into each pixel, that is, the embedding capacity is (256\times256\times6) bits in total, about 220\times220 secret image is hidden into cover image. The stego image is shown as (b) in Fig. 6, also no distortion occurs as in (a).

4.2 Analysis

4.2.1 PSNR and ROE

Generally, peak signal-to-noise ratio (PSNR) is most commonly used as a measure of the quality of the stego image in field of steganography. A larger PSNR value means that the stego image preserves the original cover image quality better. PSNR can be estimated by using following equations 5 and 6.

\[
PSNR = 20 \log\left(\frac{255}{\sqrt{MSE}}\right)
\]  

(5)

where

\[
MSE = \frac{1}{N \times M} \sum_{i=0}^{N-1} \sum_{j=1}^{M-1} \left( PV_{\text{of cover}}(i,j) - PV_{\text{of stego}}(i,j) \right)^2
\]  

(6)

\(PV\) indicates pixel value, mean squared error (MSE) is a square value of difference between the pixel value of cover image and the pixel value of stego image. Through the aforementioned formulae, we can see that PSNR value is inversely proportional to MSE value. If MSE is zero, PSNR becomes infinite sequentially, means no distortion occurs after embedding.

In this paper, we also introduce another measure criterion, ratio of embedding (ROE), to evaluate the proposed scheme. ROE illustrates the ratio that how many secret bits can be embedded into each pixel, and can be calculated using equation (7).

\[
ROE = \frac{SBL}{8}
\]  

(7)

Several 256 by 256 cover images have been tested. The PSNR and ROE are shown in Table 3. By analyzing the experimental result, it is shown that the SBL decreases progressively with NOC, and ROE decreases gradually in the meanwhile. PSNR maintains an infinite value while NOC does not exceed 128. In other words, the fewer the number of colors in the cover image, the more the secret data can be embedded, and the lower the ratio of embedding. Furthermore, no distortion of the stego image occurs after embedding except that NOC is 256. Although the proposed scheme is invalid while the NOC exceeds 128, we can eliminate the odd indices in the cover image when it has 256 colors, to reduce the NOC equal to 128. Since half of indices are excluded, it leads to a certain distortion, when the proposed scheme is applied to the degraded cover image. After that, the approximate PSNR is found to be about 51.14 in experiments, which is still superior.

<table>
<thead>
<tr>
<th>NOC</th>
<th>SBL</th>
<th>PSNR</th>
<th>ROE</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>8</td>
<td>\infty</td>
<td>1.000</td>
</tr>
<tr>
<td>2</td>
<td>7</td>
<td>\infty</td>
<td>0.875</td>
</tr>
<tr>
<td>4</td>
<td>6</td>
<td>\infty</td>
<td>0.750</td>
</tr>
<tr>
<td>8</td>
<td>5</td>
<td>\infty</td>
<td>0.625</td>
</tr>
<tr>
<td>16</td>
<td>4</td>
<td>\infty</td>
<td>0.500</td>
</tr>
<tr>
<td>32</td>
<td>3</td>
<td>\infty</td>
<td>0.375</td>
</tr>
<tr>
<td>64</td>
<td>2</td>
<td>\infty</td>
<td>0.250</td>
</tr>
<tr>
<td>128</td>
<td>1</td>
<td>\infty</td>
<td>0.125</td>
</tr>
<tr>
<td>256</td>
<td>-</td>
<td>About 51.14</td>
<td>0.125</td>
</tr>
</tbody>
</table>
4.2.2 Comparison

The proposed scheme was compared with two other methods which are based on index-color mode. ROE, SBL and distortion will be considered while illustrating the comparison. In this section, distortion is referred to as DIS for simplicity. Several representative image file formats support index-color modes, such as BMP, GIF and PNG. Particularly, BMP format only supports 1-bit, 4-bit and 8-bit color depths, PNG format supports 1-bit, 2-bit, 4-bit and 8-bit color depths, while GIF format supports from 1-bit to 8-bit color depths. For the purpose of easy comparison, and without loss of generality, 1-bit, 4-bit and 8-bit BMP images are used for comparison. As described in Section 1, EZ Stego and Fridrich’s scheme embed 1 bit of secret data into each pixel, and they are able to utilize images with 128 or more colors while the proposed scheme cannot, but the proposed scheme provides no distortion if NOC is less than or equal to 128. In Table 4, the proposed scheme cannot embed secret data when NOC is 2 in a 1-bit index-color image. However, in a 4-bit index-color image, the proposed scheme provides high embedding capacity and no distortion, with the exception of when the NOC is 16, as shown in Table 5. When the cover image is 8-bit index-color image, the proposed scheme provides no distortion after embedding, and higher ROE can be obtained than the other two schemes except when NOC is over 64 as shown in Table 6.

Table 4. 1-bit index-color image

<table>
<thead>
<tr>
<th>NOC</th>
<th>EZ Stego</th>
<th>Fridrich</th>
<th>Proposed</th>
</tr>
</thead>
<tbody>
<tr>
<td>ROE</td>
<td>SBL</td>
<td>DIS</td>
<td>ROE</td>
</tr>
<tr>
<td>1</td>
<td>Not Available</td>
<td>Not Available</td>
<td>1</td>
</tr>
<tr>
<td>2</td>
<td>1</td>
<td>Y</td>
<td>1</td>
</tr>
</tbody>
</table>

Table 5. 4-bit index-color image

<table>
<thead>
<tr>
<th>NOC</th>
<th>EZ Stego</th>
<th>Fridrich</th>
<th>Proposed</th>
</tr>
</thead>
<tbody>
<tr>
<td>ROE</td>
<td>SBL</td>
<td>DIS</td>
<td>ROE</td>
</tr>
<tr>
<td>1</td>
<td>Not Available</td>
<td>Not Available</td>
<td>1</td>
</tr>
<tr>
<td>2</td>
<td>0.25</td>
<td>1</td>
<td>Y</td>
</tr>
<tr>
<td>4</td>
<td>0.25</td>
<td>1</td>
<td>Y</td>
</tr>
<tr>
<td>8</td>
<td>0.25</td>
<td>1</td>
<td>Y</td>
</tr>
<tr>
<td>16</td>
<td>0.25</td>
<td>1</td>
<td>Y</td>
</tr>
</tbody>
</table>

Table 6. 8-bit index-color image

<table>
<thead>
<tr>
<th>NOC</th>
<th>EZ Stego</th>
<th>Fridrich</th>
<th>Proposed</th>
</tr>
</thead>
<tbody>
<tr>
<td>ROE</td>
<td>SBL</td>
<td>DIS</td>
<td>ROE</td>
</tr>
<tr>
<td>1</td>
<td>Not Available</td>
<td>Not Available</td>
<td>1</td>
</tr>
<tr>
<td>2</td>
<td>0.125</td>
<td>1</td>
<td>Y</td>
</tr>
<tr>
<td>4</td>
<td>0.125</td>
<td>1</td>
<td>Y</td>
</tr>
<tr>
<td>8</td>
<td>0.125</td>
<td>1</td>
<td>Y</td>
</tr>
<tr>
<td>16</td>
<td>0.125</td>
<td>1</td>
<td>Y</td>
</tr>
<tr>
<td>32</td>
<td>0.125</td>
<td>1</td>
<td>Y</td>
</tr>
<tr>
<td>64</td>
<td>0.125</td>
<td>1</td>
<td>Y</td>
</tr>
<tr>
<td>128</td>
<td>0.125</td>
<td>1</td>
<td>Y</td>
</tr>
<tr>
<td>256</td>
<td>0.125</td>
<td>1</td>
<td>Y</td>
</tr>
</tbody>
</table>

5 Conclusions

In this paper, we have proposed a new and efficient steganographic technique, which expands the palette and modifies the index of the cover image. The proposed scheme is based on NOC and SBL, and no distortion occurs after embedding the secret data if NOC does not exceed 128. The cover image also can be recovered without loss while extracting secret data from the stego image. Several comparisons have been taken with other previous index-color based schemes. Obviously, the proposed scheme also provides higher embedding capacity. Furthermore, the proposed scheme can be applied to not only BMP format, but also GIF or PNG format which are based on index-color technique. We can further embed more secret data when the cover image is repetitively set as a web page background, thus it has more widespread applications in today’s Internet environment.

Acknowledgements

This research was supported by 2nd Brain Korea 21 Project in 2009.

References