Location-Based Image Steganography

Amitava Nag¹, Srirupa Bhattacharyya¹, Sushanta Biswas², and Partha Pratim Sarkar²

¹Academy of Technology / Hooghly 721212 - India
²Dept. of Engineering and Technological studies, University of Kalyani / Kalyani 741 235 – India

* Corresponding Author: Amitava Nag

Received February 03, 2013; Revised March 25, 2014; Accepted April 01, 2014; Published April 30, 2014

Abstract: The objective of steganography is to conceal the presence of a secret communication. Nevertheless, with the development of steganalytic attack, also known as steganalysis, there are many statistical methods to estimate the presence and ratio of secret messages. The chi square ($\chi^2$) attack and regular-singular (RS) attack are two well recognized and widely applied statistical steganalysis schemes. In this paper, a new image steganography scheme with location selection is proposed to resist statistical steganalysis. The experiment results indicate that the proposed method can resist both $\chi^2$ attacks as well as RS attacks.

Keywords: secret communication, steganalytic attack, $\chi^2$ attack, regular-singular (RS) attack, location selection

Introduction

In recent years, the increasing popularity of digital media and communications networks has resulted in massive transmissions of secret information over the Internet. However, electronic connections between two parties are susceptible to hostile environments, and the transmission of secret information through open communication channels in the form of digital media may easily be replicated and tampered with by malicious users. Growth in different possibilities for innovative attacks by malicious users has increased the need to enforce communication security in networks. Therefore, it is necessary to use diverse methodologies to protect sensitive and secured data and resources from malicious users and to protect systems from network-based attacks. The importance of confidentiality and the integrity of sensitive data has thus led to an immense growth in the field of information hiding [1][2].

Two very important aspects of communication security are cryptography and steganography. Cryptography deals with the basic method of protecting sensitive data by rendering it unintelligible to invaders. Steganography, on the other hand, renders the message or data invisible to outsiders.

Steganography is the art of embedding secret messages into innocuous and inoffensive-looking cover documents, such as digital images [3][4][5][6][7], videos [8][10][11], audio [9][12], etc., to hide the presence of the communication. The purpose is thus to avoid drawing suspicion to the transmission of hidden information. A number of techniques have been
devised so far to embed such messages. These messages are embedded in a cover-carrier to create a stego-carrier. The process may be roughly represented as

\[ \text{Stego medium} = \text{Cover medium} + \text{Embedded message} \]

Digital images are one of the most popular cover media because they provide excellent carriers for hidden information. Different approaches for message hiding in images have been formulated, such as least significant bit (LSB) insertion methods, spread spectrum techniques, and frequency domain techniques. Many more techniques have emerged over the years, and this has led to steady growth in image steganography, in terms of both its use and its application. However, the idea of steganography has often been misused by criminals and law-breakers over the Internet, and law enforcement authorities have serious concerns about broadcasting sensitive, unwanted data over the Internet through carriers like images, audio, etc. To expose such activities, it is essential to understand the overall framework of this technology and master the methods of detecting information. Steganalysis is used for this purpose.

Steganalysis [13][2] is the technique used for detecting secret messages that were hidden using steganography without any knowledge of the embedding process. Steganalysis is what the layman describes as “breaking the security of the carrier.” Steganalysis can be broadly classified into two categories: signature steganalysis (which uses the signature of the steganography technique to detect the presence of hidden data in images) and statistical steganalysis (which uses statistics of images to uncover hidden messages). These two methods are further divided into various other approaches. However, statistical steganalysis is considered more powerful than signature steganalysis because mathematical techniques and computations are more accurate and sensitive than visual perception.

Related Work

Various steganographic techniques have been proposed to provide secure transmission of secret information using covert carriers. The simplest steganographic method is the least significant bit method [3][4][7][14], and it is used extensively in the field of information security due to its high quality and high hiding capacity. The main approach that this method uses is to embed a secret bit stream into the LSB plane of the image. LSB matching (LSBM), LSB replacement, LSB matching revised (LSMBR) [7] and LSMBR-based edge-adaptive (LSBM-EA) [14] image steganography are also popular steganographic methods similar to LSB. The LSB-replacement embedding method replaces the LSB plane of the image with the embedded message bits. However, in LSB-matching, if the embedded bits are different from the LSB of the carrier image, then the pixel value of the corresponding pixel is changed by either +1 or -1. LSBMR makes fewer changes to the cover image, and manipulates two pixels at a time, unlike LSB-replacement and LSBM, where embedding of message bits is done pixel by pixel. LSBMR is better than the other two techniques, in terms of both image distortion and steganalysis resistance. In general, the embedding positions within a cover image are chosen randomly without considering the relationship between the size of the secret message and the image content. However, LSMBR-EA uses an edge-adaptive scheme to select the embedding positions. Depending on the size of the message to be embedded, LSMBR-EA embeds the message starting from sharper edge regions to smoother edge regions. It has been shown that this technique can significantly improve security while maintaining higher visual quality of the stego-images.

Many statistical steganalysis techniques to detect the presence of embedded secret messages have also been proposed: chi square ($\chi^2$) detection [15], proposed by Westfeld and Pfitzmann, and the regular-singular (RS) attack [16], proposed by Fridrich et al., are the two most well-known and effective steganalytic techniques. The $\chi^2$ detection technique can efficiently detect sequential embedding of secret messages but fails to uncover randomly embedded messages. The RS attack technique can identify both sequentially and randomly embedded messages.

To resist steganalytic attacks, Marcal and Pereira proposed a steganographic method based on reversible histogram transformation functions (RHTFs) [17]. This technique uses RHTFs and a secret key to embed secret information into the LSB of an image. Although the RHTF algorithm can efficiently resist an RS attack, the secret key used in the process can be extracted easily by analyzing the histogram of the stego-image. Thus, the embedding rate can be estimated by an RS attack. Another drawback of the RHTF method is that the algorithm can be successfully employed only on specific cover images.

To withstand detection of embedded messages in cover images and to overcome the drawback of the RHTF method, a robust steganographic technique based on location selection is proposed in this paper.

Proposed Method

The proposed algorithm embeds messages into a grayscale image. It embeds a segment of four secret bits into a group of four cover pixels at a time. The cover image is first decomposed into non-overlapping blocks of 1x4 pixels, such that if the image is of dimension MxN, then the number of blocks generated would be (MxN)/4. The secret message to be embedded
is converted into quaternary (4-ary) numbers such that two consecutive 4-ary numbers can be embedded in each 1x4 block. Figure 1 shows the flowchart of the proposed method.

**Figure 1.** Flowchart of the proposed method: (a) embedding phase, and (b) extracting phase

**Algorithm 1: Embedding of secret message into cover image of size M×N**

1. Divide the cover image into non-overlapping blocks \( \{G_i\}_{i=1}^{T} \) of 1x4 pixels, where \( T = \frac{M \times N}{4} \) and \( M \times N \) is the dimensions of the image
2. Convert the binary sequence of the secret message to 4-ary numbers \( F \)
3. \( L = |F| \), where \( L \) represents the length or cardinality and \( L \leq 2T \)
4. Set \( i := 1 \)
5. For each \( G_i \) and two consecutive digits \( b \) and \( d \), such that \((b,d) \in F\)
   5.1 \( \{P_j\}_{j=0}^{3} \) are changed to even as \( \{P_j\}_{j=0}^{3} := (P_j)_{j=0}^{3} - (P_j)_{j=0}^{3} \mod 2 \)
   5.2 If \( b = d \)
       \( P_b := P_b + 1 \)
   5.3 Else
       a. \( (P_b, P_d) = (P_{b+1}, P_{d+1}) \)
       b. Convert \( (P_b)_{10} = (b_7 b_6 b_5 \ldots b_1 b_0)_{10} \)
          where, \( b_i \in \{0,1\} \)
Convert \((P_d)_{10} = (B_7B_6^{d-d} \ldots B_1B_0^{d-d})_2\)
where, \(B_i^d \in \{0,1\}\)

c. If \((b<d)\)
   Set \(B_i^b := 0\)

d. If \((d<b)\)
   Set \(B_i^d := 1\)

6. \(i := i + 1\)
7. If \(i \leq T\) go to step 4
8. END

Algorithm 2: Extraction of secret message from stego-image of size \(M \times N\)

1. a. If \(\{P_i\}_{i = 0, i \neq b}^{3} \) is even, then the two consecutive 4-ary digits are \((b, b)\)

   b. If \(\{P_i\}_{i = 0, i \neq b, i \neq d}^{3} \) are even, then the two consecutive 4-ary digits are \((b, d)\) where \(b < d\)

2. Convert \((P_b)_{10} = (B_7B_6^{b-b} \ldots B_1B_0^{b-b})_2\)
   If \((B_i^b = 1)\) then
      Interchange \((b, d)\) to \((d, b)\)

Considering one block of pixels and two 4-ary numbers, all the pixels of the block are changed to even. For example, if a pixel value is 103, it is converted to the even number 102. If it is already an even number, then it is left unchanged. Once all four pixels of a block are even, two consecutive 4-ary numbers from the secret message are considered. Suppose the four consecutive bits of a secret message are 0011 and, hence, their equivalent 4-ary digits are \((0, 3)\). Let the first value be denoted by \(b\) and the second value by \(d\). Hence, \(b = 0\) and \(d = 3\). So, out of the four pixels in the block, the 0th and the 3rd pixels are converted to odd by incrementing the pixel values. All the blocks of pixels of the cover image are embedded with a sequence of the secret bits following the same procedure to form the stego image. However, this embedding technique may not be sufficient to generate the secret message at the decoding side. Once the stego image reaches the recipient, the image is again converted into \(1 \times 4\) blocks of pixels on the receiver side. The index of each odd pixel must then be considered to decipher the secret message, i.e., the \(1^{st}\) and the \(2^{nd}\) pixels in a \(1 \times 4\) block are odd, then the two 4-ary numbers \((b, d)\) will be \((1, 2)\) and the secret message will comprise their binary equivalents 01 and 10. Now, there may be confusion regarding the order of consideration of these pixels (i.e., whether the binary equivalent of \(b\) is to be considered before that of \(d\), or vice versa). The secret message may either be 0110 or 1001. To solve this problem, the embedding technique is augmented with another scheme. The technique is to change the value of the second-last bit in the binary equivalent of the first odd pixel in the \(1 \times 4\) block during embedding. If the second LSB of the first odd pixel is 0, then the value of \(b\) is to be considered, followed by \(d\), to reconstruct the secret message. And if the second LSB of the first odd pixel is 1, then the value of \(d\) is to be considered, followed by \(b\).

The value of \(b\) may also be equal to the value of \(d\) during the embedding process. For example, when the secret bits to be embedded are 1111, the value of both the 4-ary numbers \(b\) and \(d\) is 3. To embed such a message, only one pixel—the \(b^{th}\) (or \(d^{th}\)) pixel—is converted to odd, and the stego image is transmitted through the covert channel. On the recipient’s side, if only one pixel is found to be odd, then the binary equivalent of its index in the \(1 \times 4\) block is considered twice to regenerate the secret message. To demonstrate the computation steps in the embedding process, we consider trivial examples for a 4-bit secret message and a cover image of size \(2 \times 2\).

Case 1:
Given: Cover image \(C = \begin{bmatrix} 106 & 105 & 102 & 103 \end{bmatrix}\)
Secret message bit sequence = 0011
Generate: \( b = 0 \) and \( d = 3 \) and
\( P_0 = 106, P_1 = 105, P_2 = 102, P_3 = 103 \)
Compute: Convert all pixels to even (by decrementing odd pixel values)
\( P_0 = 106, P_1 = 104, P_2 = 102, P_3 = 102 \)
Now, \( b=0 \) and \( d=3 \)
Hence, \( P_0 \) and \( P_3 \) are converted to odd numbers:
\( P_0 = 107 \) and \( P_3 = 103 \)
Now, \( 0<3 \), hence
\( (P_0)_2 = (107)_2 = 01101011 \), changed to: 01101001
Hence, \( P_0 = (01101001)_{10} = 105 \)
Stego: \( S = [105 \ 104 \ 102 \ 103] \)

Case 2:
Given: Cover image \( C = [106 \ 105 \ 102 \ 103] \)
Secret message bit sequence = 1101
Generate: \( b = 3 \) and \( d = 1 \) and
\( P_0 = 106, P_1 = 105, P_2 = 102, P_3 = 103 \)
Compute: Convert all pixels to even (by decrementing odd pixel values)
\( P_0 = 106, P_1 = 104, P_2 = 102, P_3 = 102 \)
Now, \( b=3 \) and \( d=1 \)
Hence, \( P_3 \) and \( P_1 \) are converted to odd numbers:
\( P_3 = 103 \) and \( P_1 = 105 \)
Now, \( 1<3 \), hence
\( (P_1)_2 = (105)_2 = 01101001 \), changed to: 01101011
Hence, \( P_1 = (01101011)_{10} = 107 \)
Stego: \( S = [106 \ 107 \ 102 \ 103] \)

Case 3:
Given: Cover image \( C = [106 \ 105 \ 102 \ 103] \)
Secret message bit sequence = 1010
Generate: \( b = 2 \) and \( d = 2 \) and
\( P_0 = 106, P_1 = 105, P_2 = 102, P_3 = 103 \)
Compute: Convert all pixels to even (by decrementing odd pixel values)
\( P_0 = 106, P_1 = 104, P_2 = 102, P_3 = 102 \)
Now, \( b = d = 2 \)
Hence, only \( P_2 \) is converted to odd:
\( P_2 = 103 \)
Stego: \( S = [106 \ 104 \ 103 \ 102] \)

To demonstrate the computation steps in the decoding process, we consider trivial examples for a stego image of size 2x2.

Case 1:
Stego: \( S = [105 \ 104 \ 102 \ 103] \)
Identify: Odd pixels: \( P_0 = 107 \) and \( P_3 = 103 \)
Here, \( 0<3 \), hence, \( b = 0 \), \( d = 3 \)
Convert: \( P_0 \) to \( (B_7^h B_6^h \ldots B_1^h B_0^h)_2 \), i.e. \( P_0 = (B_7^h B_6^0 \ldots B_1^h B_0^0)_2 : (105)_2 = 01101001 \)
Check: \( B_7^h \neq 1 \), hence, no interchange required
Hence, two consecutive 4-ary digits are (0, 3)
Secret message: 0011

Case 2:
Stego: \[ S = \begin{bmatrix} 106 & 107 & 102 & 103 \end{bmatrix} \]
Identify: Odd pixels: \( P_1 = 107 \) and \( P_3 = 103 \)
Here, \( 1 < 3 \), hence, \( b = 1 \), \( d = 3 \)
Convert: \( P_b \) to \( (B_7B_6^{b_7}B_6^{b_6}......B_1^{b_1}B_0^{b_0}) \), i.e. \( P_1 = (B_7^{b_7}B_6^{b_6}......B_1^{b_1}B_0^{b_0})_2 \):
\[ (107)_2 = 01101011 \]
Check: \( B_1^{b_1} = 1 \), hence, interchange required
Hence, two consecutive 4-ary digits are (3, 1)
Secret message: 1101

Case 3:
Stego: \[ S = \begin{bmatrix} 106 & 104 & 103 & 102 \end{bmatrix} \]
Identify: Odd pixel: \( P_2 = 103 \)
Here, \( b = d = 2 \)
Hence, two consecutive 4-ary digits are (2, 2)
Secret message: 1010

We can extend our proposed scheme to color images. A color cover image can be broken into three grayscale images corresponding to the red, green and blue planes, generating stego-images from each plane individually using the proposed location-based steganography for grayscale images. Then, the final color stego-image is generated by composing the corresponding stego-images from the red, green and blue planes. Figure 2 shows how to generate a color stego-image from one color cover image.

![Proposed embedding scheme for a color image](image)

**Figure 2.** Proposed embedding scheme for a color image

### Experiment Results

The peak signal-to-noise ratio (PSNR) is applied to compare the visual quality between cover images and stego-images. The definition of PSNR is given below:

\[
PSNR(dB) = 20 \log_{10} \frac{255}{\sqrt{MSE}}
\]

(1)
MSE is the mean squared error between the original image and the modified image, which is defined as

$$MSE = \frac{1}{M \times N} \sum_{x=1}^{M} \sum_{y=1}^{N}(I(x,y) - I'(x,y))^2$$

(2)

where M and N denote the width and height, respectively, of the cover and stego-images. Table 1 shows the PSNR values and measurement (%) of successful detection of embedded data using an RS attack on the proposed scheme for 90% embedding.

<table>
<thead>
<tr>
<th>Cover images (512×512)</th>
<th>PSNR (dB)</th>
<th>Measurement (%) of successful detection of embedded data using RS attack</th>
</tr>
</thead>
<tbody>
<tr>
<td>Airplane</td>
<td>49.6821</td>
<td>0.0223</td>
</tr>
<tr>
<td>Baboon</td>
<td>48.8917</td>
<td>-0.0070</td>
</tr>
<tr>
<td>Barbara</td>
<td>49.7701</td>
<td>-0.0095</td>
</tr>
<tr>
<td>Boat</td>
<td>50.0010</td>
<td>0.0673</td>
</tr>
<tr>
<td>Couple</td>
<td>49.6100</td>
<td>-0.0732</td>
</tr>
<tr>
<td>Goldhill</td>
<td>49.5218</td>
<td>-0.0014</td>
</tr>
<tr>
<td>Lena</td>
<td>49.3913</td>
<td>0.0740</td>
</tr>
<tr>
<td>Man</td>
<td>49.2522</td>
<td>0.0059</td>
</tr>
<tr>
<td>Peppers</td>
<td>49.0200</td>
<td>0.0160</td>
</tr>
<tr>
<td>Stream</td>
<td>50.0519</td>
<td>0.0220</td>
</tr>
</tbody>
</table>

Table 1. PSNR result for embedding the same message

![Cover Images](image1)

(a) Airplane, (b) Baboon, (c) Barbara, (d) Boat, (e) Couple, (f) Goldhill, (g) Lena, (h) Man, (i) Peppers, (j) Stream

Figure 3. Cover Images: (a) Airplane, (b) Baboon, (c) Barbara, (d) Boat, (e) Couple, (f) Goldhill, (g) Lena, (h) Man, (i) Peppers, (j) Stream

Figures 3(a) to (j) show the 512×512 grayscale cover images Airplane, Baboon, Barbara, Boat, Couple, Goldhill, Lena, Man, Peppers and Stream, and Figures 4(a) to (j) show their respective 512×512 grayscale stego-images, which are totally indistinguishable from the cover images with the naked eye.

**Statistical Attack**

Two statistical methods have been employed in this section to evaluate the security of our proposed method.

- **Resisting the χ² attack**
The $\chi^2$ attack was applied to the stego-image of Lena created with the proposed algorithm. Figure 5 shows the graph obtained after the attack was employed. The graph clearly proves the inability of the $\chi^2$ detection technique to detect the hidden message in the stego-image.

![Stego-Images](a) Airplane, (b) Baboon, (c) Barbara, (d) Boat, (e) Couple, (f) Goldhill, (g) Lena, (h) Man, (i) Peppers, (j) Stream

**Figure 4.** Stego-Images: (a) Airplane, (b) Baboon, (c) Barbara, (d) Boat, (e) Couple, (f) Goldhill, (g) Lena, (h) Man, (i) Peppers, (j) Stream

![Results of chi square attack on stego-images obtained by applying proposed algorithm](c:\new\flower\embed1.jpg:Probability of Embedding vs. Percentage Tested)

**Figure 5.** Results of chi square attack on stego-images obtained by applying proposed algorithm

### RS Attack

The RS attack was also employed on a number of stego-images generated using the proposed algorithm. Table 1 shows the percentage of hidden messages detected using the attack. It is clearly seen that the percentage of hidden messages detected is less than 1% in all the images, whereas the actual percentage of embedded messages is above 90% for most of the images.
Thus, we can conclude that the proposed method is secure against RS steganalysis.

## Merits of the Proposed Scheme

To assess the performance of the proposed scheme, comparisons between it and other related schemes [5][6][7][8][9][10] are listed in Table 2. The virtues of the proposed scheme are as follows.

**PSNR:** The average PSNR value of the proposed scheme is close to the RHTF-LSB method (decayed 0.7031 dB).

**Embedding Redundancy:** The embedding redundancy of RHTF-LSB [18] for recording the parameters is 32 bits, as shown in Table 2. In the proposed method, there is no embedding redundancy.

**Constraint in Cover Image Selection:** To improve security, cover selection is necessary in RHTF-LSB [18], which is one of the constraints of this method. In the proposed method, there are no constraints on cover image selection.

<table>
<thead>
<tr>
<th>PSNR (dB)</th>
<th>Embedding Redundancy</th>
<th>Constraints on Cover Image selection?</th>
</tr>
</thead>
<tbody>
<tr>
<td>RHTF-LSB [18]</td>
<td>50.5139</td>
<td>32 bits</td>
</tr>
<tr>
<td>Proposed Method</td>
<td>49.8108</td>
<td>0 bits</td>
</tr>
</tbody>
</table>

### Conclusion

Steganography is the artistry and skill of concealing secret messages in cover media in such a way that no one, apart from the transmitter and intended recipient, suspects the existence of the message. With steganography, the very concept is founded on the apotheosis of “security through obscurity.” In this paper, we propose an image steganographic scheme based on location selection, which is robust against both $\chi^2$ attacks and RS attacks. In addition, our proposed scheme can be applied to any image because there are no constraints on cover image selection.

### References


