Image Hiding Using Variable Length Least Significant Bits Embedding

Dr. Abbas A. Jasim
University of Basrah
College of Engineering
Computers Engineering Department

Abstract

In this work a new hiding system is proposed. It is based on Least Significant Bits (LSB) embedding of secret image into another cover image.

The proposed hiding algorithm embeds the secret image bits in the least significant bits of the cover image pixels such that the number of secret image bits that are embedded in least significant bits of cover image pixel is variable and determined randomly. Such cover image pixel may contain no secret information bit, one bit, two bits, or three bits according to the pseudo random number generator that generates integer numbers randomly between 0 and 3. The resulting image (the cover image within which the secret image is hidden) is called stego_image. Stego_image is closely related to the cover image and does not show any details of the secret information. It ensures that the eavedroppers will not have any suspicion that message bits are hidden in the image. The proposed system achieves perfect reconstruction of the secret message.
1. Introduction

Information security is becoming more important in data storage and transmission. Images are widely used in several processes. This makes protection of image data from unauthorized access to be strongly important [1]. The development of Internet provides a new way for digital information which can be made spreading faster and more conveniently. Because of the characteristic of digital images, some security problems come out besides the extensive usage of these images. [2]. Techniques and applications for information hiding have become increasingly more sophisticated and widespread as a solution of image security problem. Such applications include military and intelligence communication, covert private communication, and the protection of civilian speech against opponents [3].

Information hiding, a form of steganography, embeds data into digital media for the purpose of identification, annotation, and copyright [3]. It refers to the nearly invisible embedding of information within a host data such as text, audio, image or video [4]. The process of hiding secret information in a manner such that the existence of secret information is concealed is called stagenography [5]. It prevent outside observer from recognizing that hidden information is present. If a steganography method causes someone to suspect that there is secret information in the carrier medium, then this method fails [6]. So that hiding a message with stagenography reduces the chance of secret information being detected [7]. With high-resolution digital images as carriers (covers), detecting the presence of hidden messages has become considerably more difficult and messages embedded into an image are often imperceptible to the human eye [8][9]. Secrete data are generally embedded within insignificant area of cover image. Least significance bits of the original cover image pixels can be replaced with the secret image bits and the resultant image is not distorted. Since changing the pixel values by a small amount will not be noticeable [10].

Least significant bit (LSB) embedding is very frequently used in data hiding and there are many existing data hiding techniques to insert the secret data into the least insignificant bits (LSB) of the cover image [11][12][13]. They used LSB embedding replaces fixed number of bits of the cover image pixels with the secret information. The proposed hiding algorithm embeds the secret image bits in the least significant bits of the cover image pixels such that the random number of secret image bits to be embedded in least significant bits of cover image pixel. Matlab 2008a software is used to perform the proposed hiding system.

2. Gray Scale Images Format

Gray scale image structure represents images in a manner so called intensity image. In which image is represented as a data matrix of (M×N) elements. Each element of the matrix is corresponding to one image pixel. The elements in the intensity matrix represent various intensities, or gray levels. For eight bit pixel encoding, 0 represents black, while intensity level 255 usually represents full intensity, or white. Values within (0,255) range give the gray level of that pixel.

3. Key Generation Algorithm

Key stream is used for encryption to encrypt information. But in the proposed approach key stream is used for hiding information. It is used here as random number for allocating the secret image bits in cover image and to determine the number of bits to be embedded in each cover image pixel. The key stream is a well known in encryption especially for stream cipher. One way for its use in encryption is done by making bitwise XOR function between the plain text and the key [14].
Linear feedback shift registers (LFSRs) are widely used in key stream generators because they are well-suited for hardware implementation, produce sequences having large periods and good statistical properties. A (general) feedback shift register (FSR) of length \( L \) consists of \( L \) stages \((s_0, s_1, \ldots, s_{L-1})\) (or delay elements) numbered 0, 1, \ldots, \( L - 1 \), each capable of storing one bit. FSR having one input and one output, and a clock which controls the movement of data. During each unit of time the following operations are performed:

(i) The content of stage 0\((s_0)\) is output and forms part of the output sequence;
(ii) The content of stage \( i \) is moved to stage \( i - 1 \) for each \( i, 0 < i < L \); and
(iii) The new content of stage \( L - 1 \) is the feedback bit \( s_j = f(s_{j-1}, s_{j-2}, \ldots, s_{j-L}) \).

Hence, if the initial state of the LFSR is \([s_0, s_1, \ldots, s_{L-1}]\), then the output sequence \( s = s_0, s_1, s_2, \ldots \) is uniquely determined by the recursion: \( s_j = f(s_{j-1}, s_{j-2}, \ldots, s_{j-L}) \) for \( j > L - 1 \). The feedback function \( f \) is determined as:

\[
f = (c_1 s_{j-1} \oplus c_2 s_{j-2} \oplus \cdots \oplus c_L s_{j-L}) \quad (1)
\]

where each of the coefficients \( C_i \) is constant either 1 or 0 , \( f \) : is the feedback function which is a Boolean function, and \( s_{j-i} \) : is the previous content of stage \( L-i \).

For non zero initial function and primary feedback function, the LFSR will produce a pseudo random sequence of bits with period \( 2^{L-1} \) [12]. The block diagram of LFSR is shown in Figure 1.

4. The Proposed Hiding System

Information hiding is performed by replacing fixed number of least significant bits (LSB) of each data unit at the cover media sequentially in the scan lines across raw cover media format with the secret information bits. An attacker can recover the hidden message by repeating the process many times. The proposed hiding approach improve the hiding security by making the secret image bits to be hidden are embedded as variable length chunks of bits distributed randomly by a pseudo random number generator (PRNG) across cover image. So that cover image pixel may contain no secret information bit, one bit, two bits , or three bits according to the pseudo random number generator PRNG that generates numbers randomly between 0 and 3. In the proposed hiding system a key stream is generated and used as mask to distribute the secret image bits in the cover image and to determine the number of bits to be embedded in each cover image pixel.

4.1 The Procedure of the Proposed Hiding System

The steps of the proposed hiding approach are:

1- Generate the secrecy key using the pre defined key stream generators that are previously used for encryption. The key in this step is in the form of bit stream.
2- Decode the secret key as two bits numbers by taking each two adjacent bits and replace them by their decimal value called PRN (Pseudo Random Number). For example :

\[
K = 110010011101
\]

\[
PRN = 3 \ 0 \ 2 \ 1 \ 3 \ 1
\]

3- Arrange the resulting RNG in two dimensional form to produce Random Number Generator RNG(i,j) matrix. Such that RNG has (MxN) numbers as the same as the image. The
arrangement is implemented by take the first \( N \) numbers of 1-D RNG to form the first row of 2-D RNG then take the next \( N \) numbers of 1-D RNG to form the second row of 2-D RNG and so on.

4- Use the RNG to allocate and distribute the secret bit on the cover image. As if \( \text{RNG} (i, j) = 0 \), then no bit from the secret information will be embedded in pixel \((i, j)\) in the cover image. If \( \text{RNG} (i, j) = 1 \) then one bit of the secret message bits will be inverted and replaced with the least significant bit of pixel \((i, j)\) in the cover image. If \( \text{RNG} (i, j) = 2 \), two bits of the secret image bits will be replaced with the two least significant bits of pixel \((i, j)\) in the cover image. And if \( \text{RNG} (i, j) = 3 \), then three bits of the secret image bits will be inverted and replaced with the three least significant bits of pixel \((i, j)\) in the cover image.

5- Step 4 is repeated until all secret image bits are embedded in cover image. And the result is the stego_image that is the cover image within which the secret image is embedded.

### 4.2 The Procedure of the Reconstruction

The reconstruction steps, that reconstruct the secret image from the stego_image, are:

1- Generate the secret key using the same algorithm used in the hiding process.

2- Decode the secret key as two bits numbers by taking each two adjacent bits and replacing them by their decimal value.

3- Arrange the resulting RNG in two dimensional form to produce \( \text{RNG}(i,j) \) in the same way of hiding process.

4- Use the RNG to extract the secret information from the stego_image. As if \( \text{RNG} (i, j) = 0 \), then pixel \((i, j)\) in the cover image contain no secret information bit. If \( \text{RNG} (i, j) = 1 \), then one bit from the secret image will be extracted which is the least significant bit of pixel \((i, j)\) in the cover image. If \( \text{RNG} (i, j) = 2 \), then two bits from the secret image will be extracted which are the two least significant bits of pixel \((i, j)\) in the cover image. And if \( \text{RNG} (i, j) = 3 \), then three bits from the secret image will be extracted which are the three least significant bits of pixel \((i, j)\) in the cover image.

5- Step 4 is repeated until all secret image bits are extracted from the stego_image.

6- Rearrange the extracted bits in the original secret image form to obtain the reconstructed image by taking each eight consequent bit to form one pixel of the reconstructed image.

Figure2 shows the block diagram of the proposed hiding system.

### 5. Metrics

Several measurements are used to evaluate the performance of the proposed approach including energy, mean square error, correlation, and peak signal to noise ratio.

Image energy is used as one metric measure the similarity between the cover image and the stego_image resulting from hiding a secret image within cover image. Energy is also used to measure the similarity of the original and reconstructed secret image. The equation that was used in evaluating the energy \( E \) of a given image \( I \) of size \( M \times N \) is [9]:

\[
E = \frac{1}{MN} \sum_{r=1}^{N} \sum_{c=1}^{M} I^2(r,c) \quad \ldots (2)
\]

The Mean Square Error (MSE) is used as a metric to measure the distortion between the resulting stego_image and the original cover image and between secret and reconstructed image. MSE is evaluated as [15]:
\[ \text{MSE} = \frac{1}{M \times N} \sum_{i=1}^{N} \sum_{c=1}^{M} [I_1(r,c) - I_2(r,c)]^2 \]

Where:
MSE: Mean Square Error
\( I_1 \): is the Image,
\( I_2 \): is modified image,
And \( M \times N \): is Image size in pixel.

On the other hand, the correlation between an image \( I_1 \) and its modified version \( I_2 \) is given by the following equation [16]:

\[ \text{corr} = \frac{\sum_{i=1}^{N} \sum_{c=1}^{M} (I_1(r,c) - \bar{I}_1)(I_2(r,c) - \bar{I}_2)}{\sqrt{\left[ \sum_{i=1}^{N} \sum_{c=1}^{M} (I_1(r,c) - \bar{I}_1)^2 \right] \left[ \sum_{i=1}^{N} \sum_{c=1}^{M} (I_2(r,c) - \bar{I}_2)^2 \right]}} \]

……..(4)

where,
\( \bar{I}_1 \): image 1 mean
\( \bar{I}_2 \): image 2 mean

Another factor is used to evaluate the hiding system that is Peak Signal to noise ratio PSNR. This factor measure how the cover image get distorted by the hiding system [17]. PSNR is calculated as:

\[ \text{PSNR} = 10 \times \log_{10} \left( \frac{(2^r - 1)^2}{\text{MSE}} \right) \]

……..(5)

Where:
r: is the number of bit per pixel in each of cover image and stego_image that is 8 in this work.

7. Conclusions

The proposed image hiding system implement secret image hiding system. The secrete image bits are embedded in the least significant bits of the cover image pixels such that the number of secrete bits to be embedded in least significant bits of cover image pixel is variable and determined randomly. So that cover image pixel may contain no secrete information bit, one bit, two bits, or three bits according to the pseudo random number generator that generates numbers randomly between 0 and 3. For this reason, it provides high level of security since external eavesdropper can not estimate the location of secrete information bits nor the number of secret information bits that are embedded within any cover image pixel. The result stego_image is less distorted and is very close to the cover image so that outside observer has on suspicious that the hidden (secret) information is present. This is obvious from Table 1 results, since energy values of stego image is very close to that of cover image, MSE has very small value, correlation factor is close to 1, and large value of PSNR.

The proposed approach ensures perfect reconstruction of secret image since correlation factor between secret and reconstructed images is exactly 1, MSE between secret and reconstructed images is exactly 0, and both secret and reconstructed image have the same energy value.
8. References


Figure 1: The Block Diagram of LFSR Key Generator

Figure 2: The block diagram of the proposed image hiding System
Figure 3: Group1 Cover Image, Secret Image, stego_image and Reconstructed Image

Figure 4: Group2 Cover Image, Secret Image, stego_image and Reconstructed Image
### Table 1

<table>
<thead>
<tr>
<th>Group</th>
<th>MSE between Stego_Image and Cover Image</th>
<th>Correlation Between Stego_Image and Cover Image</th>
<th>Cover Image Energy</th>
<th>Stego_Image Energy</th>
<th>PSNR</th>
</tr>
</thead>
<tbody>
<tr>
<td>Group 1</td>
<td>8.20025e-005</td>
<td>0.998607</td>
<td>0.398423</td>
<td>0.398868</td>
<td>88.992533</td>
</tr>
<tr>
<td>Group 2</td>
<td>9.50212e-005</td>
<td>0.998857</td>
<td>0.283966</td>
<td>0.286658</td>
<td>88.352598</td>
</tr>
<tr>
<td>Group 3</td>
<td>8.07074e-005</td>
<td>0.999032</td>
<td>0.303652</td>
<td>0.303536</td>
<td>89.061670</td>
</tr>
<tr>
<td>Group 4</td>
<td>8.03988e-005</td>
<td>0.999402</td>
<td>0.238824</td>
<td>0.239183</td>
<td>89.078307</td>
</tr>
<tr>
<td>Group 5</td>
<td>8.01754e-005</td>
<td>0.998582</td>
<td>0.201385</td>
<td>0.202082</td>
<td>89.090322</td>
</tr>
</tbody>
</table>

### Table 2

<table>
<thead>
<tr>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>Group 1</td>
<td>0</td>
<td>1</td>
<td>0.420395</td>
<td>0.420395</td>
</tr>
<tr>
<td>Group 2</td>
<td>0</td>
<td>1</td>
<td>0.601370</td>
<td>0.601370</td>
</tr>
<tr>
<td>Group 3</td>
<td>0</td>
<td>1</td>
<td>0.255287</td>
<td>0.255287</td>
</tr>
<tr>
<td>Group 4</td>
<td>0</td>
<td>1</td>
<td>0.4429467</td>
<td>0.4429467</td>
</tr>
<tr>
<td>Group 5</td>
<td>0</td>
<td>1</td>
<td>0.554775</td>
<td>0.554775</td>
</tr>
</tbody>
</table>