A New Steganography Method Using Image Layers

Ömer KURTULDU, Nafiz ARICA

Computer Engineering Department
Turkish Naval Academy, Tuzla-İstanbul
okurtuldu@dho.edu.tr, narica@dho.edu.tr

Abstract

In this study we propose a new steganography method based image layers. The proposed method divides the host image into blocks and embeds the corresponding secret data bits into each block using the layers generated by the binary representation of pixel values. Given a secret bit sequence, it performs a search on the rows and columns of the layers for finding the most similar row or column. The location of row/column and its differences from the secret data is then marked by modifying minimum number of bits in the Least Significant Bits of the blocks. In the experiments, randomly selected secret messages are embedded into 100 different images. The performance of the proposed method against steganalysis techniques is evaluated using the measure of distorted pixels. The method outperforms the other steganography techniques in terms of image quality.

1. INTRODUCTION

Steganography, known as an art of hiding information, is a means of embedding data within another data while protecting its secrecy. The secret data is embedded inside the cover data by a hiding method and is sent to a receiver. The receiver applies the reverse process on the cover data and reveals the secret data. The main goal is to protect the existence of hidden data from being realized by the third parties. Thus, the methods aim to cause minimum amount of degradation on the cover data.

The methods used in image steganography can be grouped in two main categories based on the hiding domain; image domain and frequency domain methods. While image domain methods use the least significant bits in binary value of image pixel, transformation domain methods transform image data to frequency domain and perform hiding process in that domain. Image domain methods are relatively simple compared to the other methods. However, they are generally more sensitive to small changes on the image such as filtering, resizing and squeezing. The transformation domain methods, on the other hand, are more robust to changes, but their data hiding capacity is lower than the image domain methods.

In this study, we propose a new image domain method, which hides the secret data within image blocks. After representing pixel values in binary system, each image block is separated into layers, where each layer corresponds to a binary digit. The algorithm tries to match the secret data with any of the rows and columns of the blocks layers using the binary values of pixels. It searches for the most similar bit sequence and marks its location in the last layer, which contains the Least Significant Bits (LSB) of the pixels. If the most similar row/column exactly matches with the secret bit sequence, it only modifies one pixel value. If it is not an exact match, the last row of the LSB layer, which remains idle in location marking, is used to mark the differences between the most similar row/column and the secret data.

The algorithm benefits from a great number of bit sequences in rows and columns of the image layers, which constructs the search space for finding the most similar bit sequence. The large search space in each block increases the possibility of matching the secret bit sequence in the corresponding block pixels. The whole bit sequence is then embedded by changing only one bit in the LSB layer in case of an exact match. This provides minimum amount of degradation on the cover image. Performance measure of the proposed method is done against steganalysis. For this reason, randomly selected secret bit streams are embedded in 100 different cover images, then Mean Square Error (MSE) and Peak Signal-to-Noise Rate (PSNR) measures are used to measure the degradation on the cover images. In the experiments, the proposed algorithm outperforms the other image domain methods in the literature.

The paper is organized as follows. Section 2 describes the proposed algorithm by giving an example in detail. The experiments and the comparison with other studies are explained in section 4. Finally, the last section concludes the paper and discusses future studies.

3. THE PROPOSED METHOD

The method proposed in this study divides the cover image into fixed size blocks and embeds the secret bits into each block. The number of bits to be embedded in blocks is dependent on the block size. The algorithm tries to match the secret bit sequence with the rows and columns of the blocks layers using the binary values of pixels.
Each block is separated into layers using the digits in binary values of pixels. Each layer corresponds to a digit of pixels. If we consider a gray level image whose pixels are defined by a byte, the image block is separated into eight layers, where each layer contains the bits in one of eight digits. The figure-1.a shows an example image block and its corresponding layers.

Given a secret bit sequence, the algorithm applies a sequential search on the layers to find the most similar row or column. All layers except the last layer, which contains the LSBs of pixels, are used in search process. The rows and columns of the search layers create the search space for that particular image block (see figure 1.b). The LSB layer is used for marking the hiding information of the secret data.

Mathematically speaking, let’s assume that the host image, $H$ is divided into $n$ blocks, where each block $H_i$, $i=1, n$, contains $m$ number of rows and columns. Let’s also assume that binary pixel values are represented by $d$ digits. If the secret data sequence to be hidden within the block $i$ is represented as $S_i$, we can define the search process as follows:

$$\text{dist}(S_i, H_i^{j,k}) = \min_{j,k} \{ \text{dist}(S_i, H_i^{j,k}) \}$$

where $\text{dist}$ is the Manhattan distance and $H_i^{j,k}$ is the bit sequence at $k^{th}$ row/column of $j^{th}$ layer in block $i$. Since the algorithm searches both rows and columns in the layers one by one, the variable $k$ represents both rows and columns. At each value of $k$, it first searches the row $k$ and then the column $k$.

The result of this process gives the number of layer and row/column that is most similar to the secret bit sequence. The layer $p$ and the row/column $r$ are marked in the last layer $d$, which contains the LSBs. In marking, the value $(p,r)$ is considered as an index in the last layer and the $p^{th}$ row and $r^{th}$ column is changed. If the most similar sequence is a row in any layer, the marker increases the value of the pixel at $(p,r)$, otherwise it is decreased. If $\text{dist}(\cdot)$ is found as 0, which means an exact match, we can generate the block $H'_i$ of stego-image as
However, if $\text{dist}(\cdot) > 0$ then, the last row of block is also modified by marking the index of unmatched bits in the secret bit sequence. First, the indexes of secret bits $S_i(t)$ which are not equal to the most similar row/column $H_r^{p,r}(t)$ are found by;

$$unmatched = \text{noteq}(S_i(t),H_r^{p,r}(t)) \quad t = 1...m$$

Then these indexes are marked by changing 0s and 1s in the last row $H_i'(d,\ast)$:

$$H_i'(d,unmatched) = H_i(d,unmatched)$$

In (2), it should be noted that the pixel values drastically changes if their values 0 or 255. In order to prevent the huge amount of degradation, if the pixel value is 255 the corresponding $r^{th}$ row of layer $p$ is omitted and the exact match is searched only in $p^{th}$ column of that layer. In case the value is 0, the $r^{th}$ column is omitted. As a result, the search space is decreased by one for each value of 0 or 255.

When extracting the secret bits in a stego-image, each image block is examined and the differences between the host image and stego-image are located. The degraded pixel in the rows other than the last one gives the location of secret bit sequence. If there is no degradation in the last row, then the secret data is directly extracted using the location information. Otherwise, the indexes of degraded pixels in the last row are used for converting the most similar row/column into the secret data.

Let's explain the above process with a simple example. Assume that we have a host image block with 4x4 pixels and a secret data with 4 bits given as;

$$H_i = \begin{bmatrix} 12 & 10 & 10 & 9 \\ 11 & 9 & 8 & 7 \\ 1100 & 1010 & 1010 & 1001 \\ 1011 & 1001 & 1001 & 1000 \\ 1011 & 1001 & 1000 & 0111 \\ 1100 & 1010 & 1010 & 1010 \end{bmatrix}$$

$S_i = [0101]$

For the sake of simplicity, the pixels are represented by 4 digits and take values between 0 and 15. The layers of this particular block are represented as;

$$\begin{bmatrix} 1111 & 1 & 0 & 0 & 0 \\ 1111 & 1 & 0 & 0 & 0 \\ 1111 & 0 & 0 & 0 & 1 \\ 1111 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 \end{bmatrix}$$

Layer 1  Layer 2

$$\begin{bmatrix} 0 & 1 & 1 & 0 & 0 & 0 & 0 & 1 \ 1 & 0 & 0 & 0 & 1 & 1 & 1 & 0 \ 1 & 0 & 0 & 1 & 1 & 1 & 0 & 1 \ 0 & 1 & 1 & 1 & 0 & 0 & 0 & 0 \end{bmatrix}$$

Layer 3  Layer 4

The algorithm searches for the secret bit sequence $[0 1 0 1]$ in the first three layers. It finds the most similar sequence at the 3rd row of the 2nd layer. In other words, the result of equation (1) is (2,3). It then marks the pixel value at the index (2,3). Since the match is found in a row, the pixel value 9 is increased to 10. In addition, the unmatched bit of secret data is found at the 2nd index. This is marked in the 2nd element of 4th row. Then stego-image block $H_i'$ becomes;

$$H_i' = \begin{bmatrix} 12 & 10 & 10 & 9 \\ 11 & 9 & 10 & 8 \\ 11 & 9 & 8 & 7 \\ 12 & 11 & 10 & 10 \end{bmatrix}$$

In recovering the secret message, the stego-image is compared with the host image block by block and the degradation at the index (2,3) for this particular block is recognized. The bit sequence $[0 0 0 1]$ in the third row of the second layer is extracted. The last row of layer 4 is checked and the modification in the 2nd index is recognized. Then the 2nd index of $[0 0 0 1]$ is converted and the secret data $[0 1 0 1]$ is found.

4. EXPERIMENTS

The performance of steganography methods is mainly measured by the degree of robustness to the steganalysis techniques. The first step in steganalysis is to determine the existence of secret message in host image by measuring the degradation in the stego-image.

Two of the error metrics used to compare the host image and stego-image are Mean Square Error (MSE) and Peak Signal to Noise Ratio (PSNR);

$$MSE = \frac{\sum_{m,n}(H(m,n) - H'(m,n))^2}{MN}$$

$$PSNR = 10 \log_{10} \left( \frac{255^2}{MSE} \right)$$

where $M$ and $N$ are the size of the images.
The performance of proposed method is evaluated using 100 grey level images with length 256x256. In the experiments, randomly selected messages are embedded into all the images. The images are divided into 8 by 8 blocks and 8 bit secret data is embedded into each block. Considering the size of the images and the pixel values represented by 8 bits, the length of messages embedded into each image becomes 8192 bits.

In order to compare the method with the other methods in the literature, the same hiding experiment is performed using various methods [1], [2], [3], [4], [5]. The comparison results are given in Table-1, which shows the average values in 100 different images and secret messages.

As we discussed in section 2, LSB matching revisited method performs the highest stego-image quality among all the other available methods. On the other hand, the proposed method outperforms LSB matching and the other methods in our experiments in terms of MSE and PSNR metrics.

Table 1: Comparison of the proposed method with the other image domain methods

<table>
<thead>
<tr>
<th>Methods</th>
<th>MSE</th>
<th>PSNR (db)</th>
</tr>
</thead>
<tbody>
<tr>
<td>LSB Replacement [1]</td>
<td>0.028402</td>
<td>63.63</td>
</tr>
<tr>
<td>A New Algorithm for Hiding Gray Images Using Blocks [4]</td>
<td>0.407495</td>
<td>52.06</td>
</tr>
<tr>
<td>Hiding Secret Message in Edges of the Image [3]</td>
<td>0.029459</td>
<td>63.47</td>
</tr>
<tr>
<td>An Architecture Platform for Grey Level Modification Steganography [5]</td>
<td>0.029875</td>
<td>63.41</td>
</tr>
<tr>
<td>LSB Matching Revisited [2]</td>
<td>0.021988</td>
<td>64.74</td>
</tr>
<tr>
<td>Proposed Method</td>
<td>0.012994</td>
<td>67.03</td>
</tr>
</tbody>
</table>

In the experiments, the exact match is found in 21% of the blocks. It means that 8 bit secret data is hidden using only one bit modification. In the average, 1.76 bits are modified in order to hide 8 bit data. If we compare our method with LSB matching revisited method, while our method modifies 0.44 bit, LSB matching revisited modifies 1 bit to embed 2 bits. As a result the proposed method decreases the degradation in the quality of the host image.

It should be noted that our method embeds one byte data into an 8 by 8 block. While LSB matching revisited method tries to embed one bit data in every pixel, one bit secret data is embedded into eight host pixels in our method. One might claim that the reason for higher performance is because of this lower hiding capacity. However, as we discussed above, the hiding capacity is not as important as image quality because of short message sizes and variable size/number of host images. Therefore the hiding capacity of our method is sufficient for steganography applications.

5. CONCLUSION & FUTURE STUDIES

While almost all the image domain techniques focus on only the least significant bits of the pixel values, the proposed method tries to find a match in the higher layers of host image. In LSB based strategy, the best image quality is performed in LSB matching revisited method, which modifies at least one pixel value for two secret bits. On the other hand, our method benefits from the possibility of matching whole secret bit sequence with any row and column of the image block. The rows and columns in the higher layers constructs a large search space for secret bit sequence to be matched in each block. If we consider a block size with 8 and a gray level host image, 8 bit secret data can be embedded into a block with a modification of one value.

The main constraint in our method is to hide one secret bit sequence into each block. However, we can spread the search process in the whole image and find a correspondence between the secret bit sequences and the search layers of all blocks in the image. This approach makes it possible to embed more than one bit sequence into each block. Then the message size that can be embedded can be increased or decreased. In addition, it allows us to hide different number of sequences into the blocks according to the optimal correspondence. Therefore, the quality of stego-image can increase.

In our future study, we plan to do research on variable size secret data hiding methods.

6. REFERENCES