IJARCCE

ISSN (Online) 2278-1021
ISSN (Print) 2319 5940

International Journal of Advanced Research in Computer and Communication Engineering
ISO 3297:2007 Certified
Vol. 5, Issue 10, October 2016

Random Area Selective Image Steganography with LSBMR

Tibin Thomas[1], Neethan Elizabeth Abraham[2]

Assistant Professor, Department of CSE, Saintgits College of Engineering, Kottayam, Kerala, India [1]

Assistant Professor, Department of ECE, Mangalam College of Engineering, Kottayam, Kerala, India [2]

Abstract: Steganography is the way of hiding the information in a multimedia. So no one other than the creator and the receiver can identify the existence of the message. Even though there are so many image steganography exist, most of them are not adapted to the amount of secret data that to be embedded. In this paper, we are presenting a method which dynamically selects the region for embedding the data based on its size. We use a selection method to find the region for embedding and in that region, we embed the message using the LSBMR method. Regions may be a single pixel or group of pixels. Regions are scattered in different locations of the image and thus it achieves high security.

Keywords: Substitution Encryption Method, Steganography, LSB, LSBM, LSBMR, Steganalysis

I. INTRODUCTION

The objective of using steganography is to hide the secret data in a covering object like audio, video and image etc. Steganography mainly works by replacing the bits in the image with the bits in the message such that the replacement should not tamper the image. The methods which are used to find the hidden data in an image are called steganalysis algorithms. A lot of steganalysis methods are existing. If a particular steganalysis can find the presence of secret data in the image, then that stago system is said to be compromised. Most of the steganalysis work by looking the structural changes happened in the image. As the data need to embed is large then the structural change is also being large. So stego systems should achieve two properties 1) Un-Detectability and 2) amount of data that can be embedded

A. Organization of the paper

The remaining of the paper is organized as below. In the next section we give a small introduction to the related works. It describes the some of the currently existing steganography techniques. In the section III we explain the working of LSBM Revisited algorithm and in section IV we explain the proposed system and how it improves the security of secret data.

II. SURVEY

The simple method is an LSB replacement method [1]. It replaces the least bit of each pixel value in the image with a bit in the secret data. This simplest scheme introduces some changes in the structure of the image and thus it was very easy to find the existence of secret data using the some steganalysis tools like Chi-squared attack [2] and sample pair analysis [3].The LSB Matching algorithm is another method which increment/decrement the value of pixel by one if the data bit and LSB bit are not matching. However, the LSB matching can still be vulnerable to the histogram attack [5].Also, many other stenographic algorithms proposed to analyze the LSBM algorithm [6-9]. The LSBM Revisited [4] use pairs of two pixelswhose values are adjusted to store two secret bits. Actually the first pixel carries the first bit and the second bit is stored/retrieved by the combination of two pixels. Even though the above methods can hide the data in the image, they have poor security performance.

III. LEAST SIGNIFICANT BIT MATCHING REVISITED

Here we are describing the working of LSBMR [4] technique in detail. As said above the LSBMR choose a pair of bits from the data and hides one bit in a pixel and second as the relationship between two pixels. First of all LSBMR technique group bits in the secret data and pixels in the image as pairs of two, i.e.If x_1, and $y_1$are the values of a particular component (any one of RGBcomponents)of two pixels and m_1 and m_2 are the bits to embed. Then the algorithm changes the value of the pixel as x_2 and $y_2$such that

$$\text{LSB}(x_2) = m_1 \text{and LSB}\left(\left\lfloor\frac{x2}{2}\right\rfloor + y2\right) = m_2 \qquad (1)$$

By using these relationship LSBMR got less modification rate for the image as compared to the LSB and LSB matching schemes.

The value of a colour of pixel always ranges from 0…. 225. In order to embed two bits we need to make the LSB of a colour in pair of pixel as it satisfies the above equation. So in LSBMR technique we have four cases for embedding which can be chosen based on the current value of the hiding bits. Based on the selected case the value of the current pixel (x_1, y_1) changes to (x_2, y_2)

IJARCCE

ISSN (Online) 2278-1021
ISSN (Print) 2319 5940

International Journal of Advanced Research in Computer and Communication Engineering
ISO 3297:2007 Certified
Vol. 5, Issue 10, October 2016

Case a: if currently

$LSB(x_1) = m_1$ and $f(x_1, y_1) = m_2$

Then $(x_2, y_2) = (x_1, y_1)$

Case b: if currently

$LSB(x_1) = m_1$ and $f(x_1, y_1) \neq m_2$

Then $(x_2, y_2) = (x_1, y_1+r)$

Case c: if currently

$LSB(x_1) \neq m_1$ and $f(x_1, y_1) = m_2$

Then $(x_2, y_2) = (x_1-1, y_1)$

Case d: if currently

$LSB(x_1) \neq m_1$ and $f(x_1, y_1) \neq m_2$

Then $(x_2, y_2) = (x_1+1, y_1)$

Where

$$f(x_1, y_1) = LSB\left(\left\lfloor \frac{x1}{2} \right\rfloor + y1\right)$$

The receiver can extract the image using the equation 1.

IV. PROPOSED SYSTEM

The proposed system uses the above mentioned LSBMR scheme with some initial mechanisms to improve the security performance. For improving the security these systems use two private keys (known only to the two communicating entities) which are used to hide/unhide the data. The first key is used to scatter the portions of the cover image into different areas and second key is used to select the area at which each bit is to be embedded.

The flow diagram of the proposed system is shown in Fig. 1: data embedding and Fig 2: Data extraction. Each image should divided into different small regions called TILE. Each tile is of size n x n pixels and the TILE's size may vary from one pixel to largest square matrix of pixels possible inside the image. The parameter n is initialized from one. Values of n varies upon the amount of secret bits to hide. In order for the correct extraction of the message at the receiver side, we also need to embed these type of parameters into the image and we called it as SECOND data..

A. Region Selection

As said above we start with tile of size 1x1 pixel. The LSBMR technique require pairs of two pixels to embed two bits of secret data. So to make pair we select the first pixel from the first tile and the second pixel from the second tile. To make the next pair of pixel we choose the third and fourth tile and so on. These tiles are selected based on the value of key1. The system first generates the binary value of key1. and then the left topmost tile is considered as the first tile.For selecting the next tile it checks the left most bit (b0) of key1. If it is 1 we select the right tile of the current as the next tile or of it is zero we select the bottom tile of the current tile as the next one. For selecting the third tile system uses the value of second left most bits of key1 (b1). This way, using the value of key1 (like b0, b1, b2, b3…. bn) system selects the tiles until it reaches the rightmost or bottommost tile. If the size of the key1 is not big enough to reach the right/bottom tile, then repeatedly append the key1 with itself until it reach enough size, i.e.

b0,b1,b2,b3….bnb0,b1,b2,b3,….bnb0,b1,b2,b3,….bn

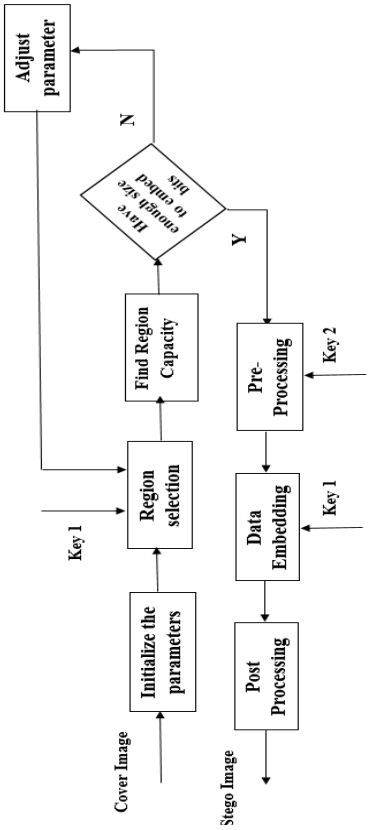


Fig 1: Flow diagram of Message Embedding

On reaching the right or bottom end check whether the pair of selected tiles (here a single pixel is a tile) is enough to embed the secret image. If they are enough, then using the LSBMR method to embed the bits. If the pairs are not enough to embed the bits, increase the size of each tile i.e to 2x2 size.The systemfirst pair the pixels in the first tile and then select the next 2x2 tile based on the value of key_1. Check for the selected region's capacity and if it not enough adjust the parameter and do the same until we reach n x n size tiles which is enough to hide the entire secret data. To get even number of pixels in a tile only even numbers are assigned for size n. These region selection stage is only to find the size of the tile. Actual region for each bit is selected only after the preprocessing stage. The tile size is embedded in the post processing stage of the system, which can be used by the receiver.

B. Preprocessing stage

To enhance the security the data are scattered in different places. Even though the region selection based on key1 scatter the bits in different places, to achieve more security, we pass the cover image into a preprocessing stage before data embedding

DOI 10.17148/IJARCCE.2016.51003

IJARCCE

ISSN (Online) 2278-1021
ISSN (Print) 2319 5940

International Journal of Advanced Research in Computer and Communication Engineering
ISO 3297:2007 Certified
Vol. 5, Issue 10, October 2016

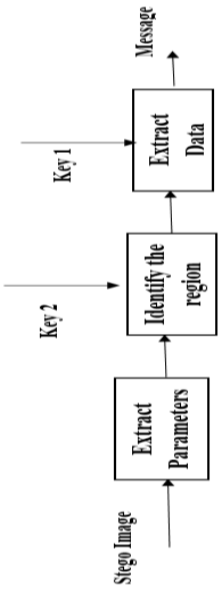
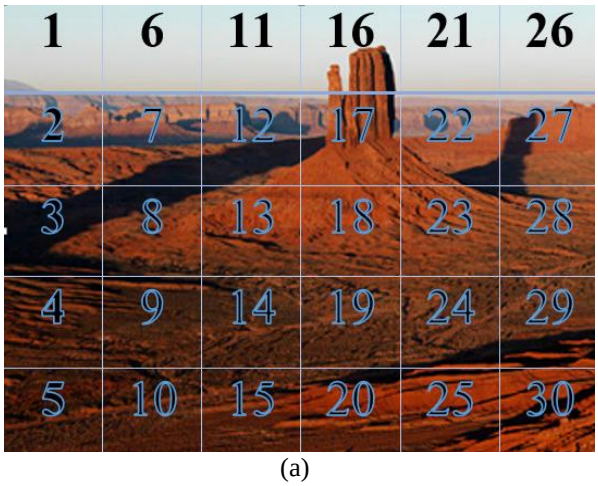


Fig 2: Flow diagram of Message Extraction

At this stage we consider the n x n size tiled cover image, i.e. the image passed out the preprocessing stage which identified the required tile size. In this system instead of scattering the secret bits into a different location of the image, here the system displaces the tiles in the image into different locations and then embed the secret bits.The tiles are displaced from top to bottom in a circular pattern. The number of positions each tile to be moved is determined using a function that uses key_2 as the parameter. After the displacement of the TILEs the bits are embedded and then do the reverse displacement. Since the number of locations to be moved depends on key_2, only the person who knows the key2 can extract the secret bits. To increase much more security each column of the n x n image can be displaced into a varying number of times. The below figure 3 (a) and (b) shows an example of the cover image and the preprocessed cover image respectively. Here the function with parameter key_2 generate the value One. So the tiles in the first column circulated once, the second column rotated for two times, third column rotated for three times and so on.



(a)

(b)

Fig 3: (a) Original cover image (b) Displayed cover image, where each column displaced in varying amount.

C. Data Embedding

After the preprocessing stage adispersed cover image is obtained which is used for data embedding. Now with the help of key_1 we find the pair of pixels again and with LSBMR technique we embed the bits into it.

D. Post processing

After the embedding stage the scattered cover image should be re-rotated to make it as an original cover image. We also embed some parameters like size of the tile (n) (can be in an encrypted format using key_1 OR key_2), the component of the pixel which is used for embedding etc. as SECOND data.

V. EVALUATION

The proposed system is simulated in Qt platform.

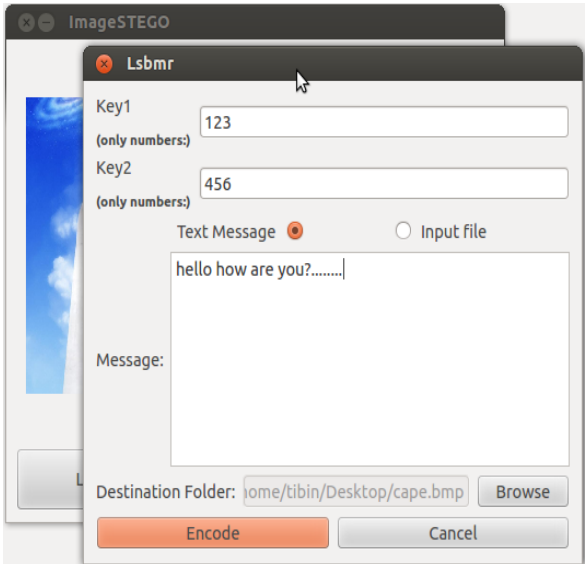
A. Simulation

Fig 4 (a) and (b) shows the home screen and keys uploading screen of our system. In the created system the key_1 and key_2 are assumed to be shared between the two communicating entities in a secured manner. To meet the bottom or right end of the cover image while choosing the region for embedding, need a key of enough size. So in the created system we append the input key1 with itself until it becomes 2kbits size.

(a)

ISSN (Online) 2278-1021
ISSN (Print) 2319 5940

International Journal of Advanced Research in Computer and Communication Engineering
ISO 3297:2007 Certified
Vol. 5, Issue 10, October 2016



(b)

Fig 4 (a) Home Screen (b) Keys uploading

The system has the option for encoding as well as decoding the secret data from the image. The secret message can be given either the text message in the given box or as file of any type. For correctly extracting the information at the receiver the SECOND data are stored in the image.

We used the first five rows of the pixel to embed such type of parameters and embedded it with LSB method on green and blue. So while making the tiles we neglect those rows. Table 1 shows the parameters and the number of bits used for that.

TABLE 1: PARAMETERS EMBEDDED

Parameters	No of bits used
Tile size	20 bits
Pixel component used	2bits : 00→RED 01→BLUE 10→ GREEN
Text or File	1bit: 0→Text 1→File
File name size	4bits (Gives number of character in the file name)
Filename	128bits
File type size	3bits (Gives number of character in the file type)
File Type	64bits

B. Analysis

n order to analyze the stenographic system we used the chi-square steganalysis tool. The Fig 4 gives the analysis reports for the inputted cover image (before embedding), the stego image width.7kb data, and 1.5KB data. The application is tested using different images like PNG, BMP, TIFF, XPM, etc. Plain text with different size are tested. Similarly, files of type. Text, .Pdf, .Ps, and image files are tested. For all these testing the cover image and stego image didn't show dissimilarities.

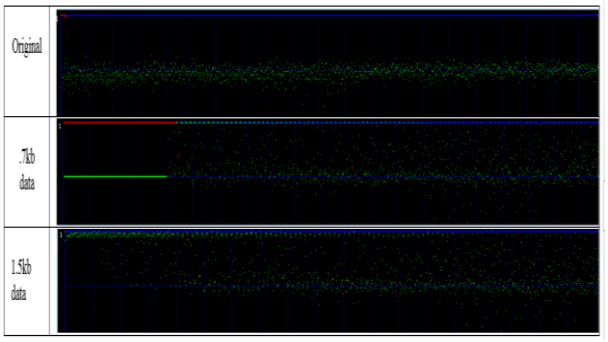


Fig 4 : Chi-square analysis

VI. CONCLUSION

Here we developed an image steganography system which uses some secret bits scattering technique with the LSBMR embedding method. As we saw the shifting of the tiles into different region will mover successive bits of secret message at different location of the cover image. Thus the generated hidden data in stego image will be difficult to detect as compared to the existing system. As a future expansion, we like to create the system which uses same methods, but on audio/ video files as the cover media.

REFERENCES

[1] W. Bender, D. Gruhl, N. Morimoto, and A. Lu, Techniques for data hiding, IBM System Journal, vol. 35, no. 3, pp. 313-336, 1996.
[2] A. Westfeld and A. Pfitzmann, "Attacks on steganographic systems," in Proc. 3rd Int. Workshop on Information Hiding, 1999, vol. 1768, pp. 61–76.
[3] S. Dumitrescu, X.Wu, and Z.Wang, "Detection of LSB steganographyvia sample pair analysis," IEEE Trans. Signal Process., vol. 51, no. 7, pp. 1995–2007, Jul. 2003.
[4] J. Mielikainen, "LSB matching revisited," IEEE Signal Process. Lett.,vol. 13, no. 5, pp. 285–287, May 2006.
[5] Cox, G. Derr, "Steganalysis for LSB matching in images with high-frequency noise," in Proc. IEEE Workshop on Multimedia Signal Processing, Vol. 34, PP. 385--388, 2007.
[6] J. Harmsen and W. Pearlman, "Steganalysis of additive-noise modelable information hiding," Proc. SPIE Electronic Imaging, vol. 5020, pp. 131–142, 2003.
[7] X. Li, T. Zeng, and B. Yang, "Detecting LSB matching by applying calibration technique for difference image," in Proc. 10th ACM Workshop on Multimedia and Security, Oxford, U.K., 2008, pp. 133–138.
[8] F. Huang, B. Li, and J. Huang, "Attack LSB matching steganography by counting alteration rate of the number of neighbourhood gray levels," in Proc. IEEE Int. Conf. Image Processing, Oct. 16–19, 2007, vol. 1, pp. 401–404.
[9] A. D. Ker, "Steganalysis of LSB matching in grayscale images," IEEE Signal Process. Lett., vol. 12, no. 6, pp. 441–444, Jun. 2005.

DOI 10.17148/IJARCCE.2016.51003