Improved LSB image steganography using MSB filtering of pixels

Abdul Jamsheed V
National Institute of Technology
Tiruchirappalli, TamilNadu, India-620015
205219001@nitt.edu

Dr. B Janet
National Institute of Technology
Tiruchirappalli, TamilNadu, India -620015
janet@nitt.edu

Abstract: Steganography is the technique in which private or sensitive messages are hidden into something such as image, video or audio. It involves hiding text so it appears to be a normal image or other media file. The attackers doesn't have any idea that image contains some sensitive information and which algorithm can be used to extract it. This paper is focused on bitmap image which is widely used for LSB steganography than any other image format. This is an improved version of existing LSB based image steganography. Total pixels are classified based on the MSB bit filtering into dark pixels and light pixels before encoding and any one type of the pixels are used to hide message. It is found that the proposed technique will provide additional security to the normal image steganography and can be able to hide large data in single image.

Keywords*—Image Steganography, Filtering Algorithm, hiding of Message, Light and dark pixel Filtering, LSB Image Steganography.*

I. INTRODUCTION

A large amount of sensitive information is lost every year by infiltrators during the transmission of data. Many encryption methods are widely used in order to encrypt or decrypt the data. And this is not enough, hiding the data itself is required to protect the sensitive information from the intruders.

Steganography is the process hiding secret message or sensitive messages in carrier such as text, voice, image or video. Digital images are the common carriers used for steganographic technique due to their popularity in the internet and high data transmission capacity without degrading the image quality. Image steganography is more secure than text and voice steganography and efficient than the video steganography.

In the computer point of view, an image is a bundle of pixels that have different light intensities in different parts of the image. These pixels are displayed horizontally row by row. The best known steganographic method is LSB steganography, which replaces Least Significant Bit of the pixel and stores the secret information. Many improved version of this algorithm has already been implemented, but no good version of this adds security to the existing LSB method. Stego image is the image which contains the secret message and cover image is the image which is used to hide the secret information.

Here, we improve default LSB technique by filtering the pixels based on MSB. We classify the pixels into Dark pixel or light pixel and hide the sensitive information only in either dark pixel or light pixel. Thereby improving the security and if anyone tries to get the message by only taking LSB bits, he/she will fail to get the original message.

II. PREVIOUS WORKS

Different Steganographic techniques are implemented from very old times on variety of electronic mediums. A method proposed by Qing X and Yunhua states that information can

be hidden in all RGB planes based on human visual system [5]. Masud Karim and M.I. Hossain proposed a new innovative approach based on LSB using secret key. The secret key is encrypted and the hidden information is stored into different position of LSB of image [2]. The method proposed by R. J. Chen et al presents the novel multi bit bitwise adaptive embedding algorithm for data hiding by evaluating the most similar value to replace the original one [6]. But these techniques didn't talk about the security of the secret message and how the pixel classification can be used to hide the data.

III. PROPOSED TECHNIQUE

In this method we use colour image. A colour pixel can be represented as a mixture of R, G and B proportions. Each colour is represented by a Byte or 8 bits, so total of 3 bytes or 24 bits are required to represent a pixel. Thus image is an array of many bytes, each byte represents a single colour information. In proposed technique each byte is used to store a bit of message. In this steganographic method, a MSB based filtering algorithm is used to hide information. The Most Significant Bit specify where to hide the secret message. User will specify whether the secret information should be added in darker pixel or lighter pixel. Lighter pixel means MSB bits of R (Red), G (Green), B (Blue) component of pixel contain at least 2 bits 1's and darker pixel means MSB bits of R(Red), G (Green) and B (Blue) component of pixel contains at least 2 0's. Below diagram shows how dark and light pixels are distinguished using the MSB bit.

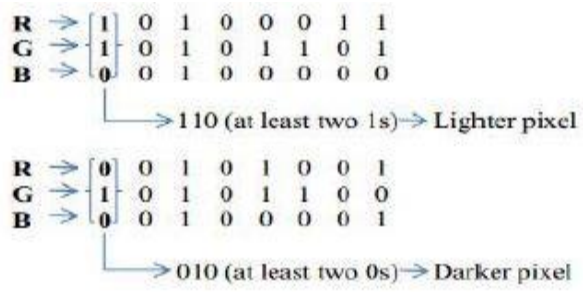

Figure 1. Concept of dark and light pixel

A. EMBEDDING PROCESS

In a digital bitmap image, colours are arranged byte by byte as R-G-B-R-G-B-R-G-B… etc. In this method, a message bit is injected in the LSB of every byte. Using the concept of light and dark pixel, we choose either one of the pixel to store the data. This decision is provided by the user. Once we select the pixel type we will embed the data only in these type of pixel in the whole image. Every data byte is converted to its 8-bit binary code using ASCII values. A pixel contains 3 bytes. To store a byte of data we need 3 pixels, each of which having 3 bytes. So total of 9 bytes. The LSB bit of first 8 bytes are used to store the binary data. If the value of the pixel byte is made odd if 1 occurs and made even if message bit is even.

Here we consider an example to explain the above concept, suppose we want to hide a message '*Hi*' in the cover image for steganography. Since the message is of 2-bytes, we need 2 x 3 = 6 pixels to encode the data in the corresponding image. We can store our message in either light pixel or dark pixel. Here we choose light pixel to store the secret message. So there won't be any data hidden in the darker pixels. Take a sample image of 4 x 3 with total of 12-pixels, which is enough to encode the given data. But this may not be true for every case. If the image contains more number of darker pixels and the lighter pixels are low, we need to ensure that there are enough light pixels to embed our message. Usually this doesn't make any problem if we use high density pixel images.

```
[(27, 64, 164), (248, 244, 194),
(174, 246, 250), (149, 95, 232),

(188, 156, 169), (71, 167, 127),
(132, 173, 97), (113, 69, 206),

(255, 29, 213), (53, 153, 220),
(246, 225, 229), (142, 82, 175)]
```

ASCII value of the alphabet '**H** ' is 72 whose binary equivalent is 01001000. First pixel is a darker pixel, we will skip darker pixel, as we are only hiding the data in lighter pixel. Taking

the next 3-pixels (248, 244, 194), (174, 246, 250), (149, 95, 232) to encode. Now change the pixel value to odd by subtracting 1(if it is not odd) for message bit 1 and change it to even by subtracting 1 (if it is not even) for message bit 0. So, the modified pixels are (248,243,194), (174,245,250), (148, 94, 232). Since we have to encode more data, we will keep the last byte of the third pixel to even. If we don't have to store any more data we simply make the last byte as odd. Similarly, '**i**' can also be encoded in this image. For the simplicity we used *HI* message. We can embed large data into the image. The efficiency of this will depend upon the quality of the image as well.

The pixel representation of new image will look like:

```
[(27, 64, 164), (248, 243, 194)
(174, 245, 250), (148, 94, 232)

(188, 155, 167), (70, 167, 126)
(132, 173, 96), (112, 69, 205),

(254, 29, 213), (53, 153, 220),
(246, 225, 229), (142, 82, 175)
```

I. EMBEDDING ALGORITHM

1. Get the message to be encoded into the image.
2. Convert the message into binary number.
3. Get the cover image for embedding the secret message.
4. Get the user prompt, whether to hide data in lighter pixel or darker pixel.
5. Collect the MSB bits from the pixels to identify whether it is dark pixel or light pixel (Red, Green, Blue colour component).
6. If the MSB bits of the pixel contain 2 or more 1's it is classified as lighter pixel and MSB bits of the pixel contain 2 or more 0's, it is classified as darker pixel. Select the user specified pixel for hiding the data. Otherwise skip the pixel.
7. If the data message bit is 1 and LSB bit is odd, skip the byte. If the LSB bit is even, subtract 1 from the binary value of byte and make it odd.
8. If the data message bit is 0 and LSB bit is even, skip the byte. If the LSB bit is odd, subtract 1 from the binary value of byte and make it even.

B. EXTRACTING PROCESS

First of all collect the series of bytes of the Stego image and arrange it into light or dark pixel based on the user need. Decode the message from pixel either from dark or light pixel, which will be known only to sender and receiver. Collect 3 bytes and verify the type of the pixel. If it is the pixel which we used to encode, then decode the message from the pixel. While decoding, three pixels are taken at a time, till the last value is odd, if last value is odd means end of the message. Every 3-pixels contain a binary data. The binary data is extracted by the same encoding logic. If the value is odd we extract the binary bit 1 and append into message value else 0 is appended into message value.

I. EXTRACTING ALGORITHM

1. Get the output image.
2. Select the type of pixel from which we want to extract the data.

3. Collect the MSB bits from the pixels to identify whether it is dark pixel or light pixel (Red, Green, Blue colour component).

4. If the MSB bits of the pixel contain 2 or more 1's it is classified as lighter pixel and MSB bits of the pixel contain 2 or more 0's, it is classified as darker pixel. Select the user specified pixel for extracting the data. Otherwise skip the pixel.

5. If the decimal value of the byte is even, extract 0. If the decimal value of the byte is odd, extract 1.

6. Stop decoding if the last bit of the 3rd pixel is odd.

7. Convert the decoded bits into character

IV. EXPERIMENTAL RESULT

A bitmap image named Lena.png and Peppers.png are used as the cover image as shown in the figure. 2 The outputs of the experiments are remarkably similar to original image. The secret message hidden in the image is "*India planned strategic and military relations with United States of America to counter Pakistan, this will bring peace in Jammu Kashmir. This may also called as a strategic plan to gain the vote in the coming elections by injecting the patriotism which always works in the Indian politics".* The given message is just a sample message and doesn't have anything to do with the steganographic technique or the filtering algorithm. The histograms are also shown below (in figure. 3). It is seen that proposed steganographic method does not change much in the histogram and the changes cannot be detected in the naked eye or normal zoom of the cover image. This is because the data is not embedded in the consecutive pixels but it is injected into random pixels.

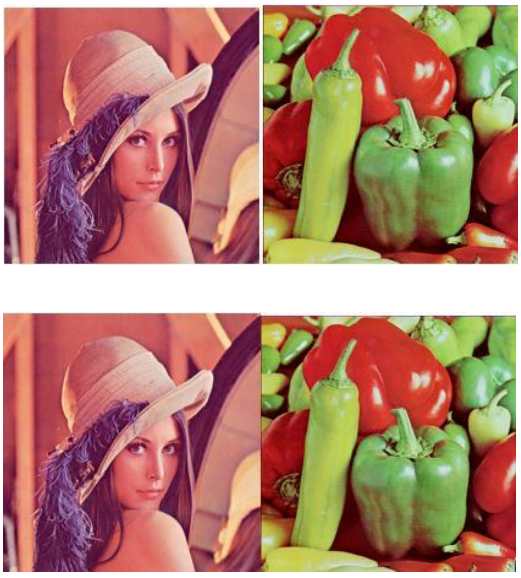

Figure 2. The image before embedding the secret info (top) and image after embedding the secret info (bottom) of the steganographic process.

Figure 3. The histogram of Lena image without embedding the secret message (left) and the corresponding histogram of image in which secret information is embedded (right)

V. CONCLUSION

In the above method of hiding data, we have introduced a new concept of Steganography in which random pixels are used to store the message data. This random pixels are selected by MSB filtering of the pixels. We would like to highlight that the goal of the technique is not to just hide the data but to add additional security to make it difficult to the unauthorized people to identify the presence of a secret message. In conventional LSB Steganography technique only least significant bit is replaced and at the end fifteen 1's are added to know the end of message. This can be easily exploited by the intruders to extract the original message. Here we use MSB bit to filter the pixel and either dark pixel or lighter pixel are used to hide the message. So the proposed technique of steganography fulfils the criteria and adds a special security to the sensitive information.

REFERENCES

[1] T. Morkel, J. H. P. Eloff, and M. S. Olivier, "An overview of image steganography" *New Knowledge Today Conference*,
Sandton, pp. 1-11, 2005.

[2] S.M. M. Karim, M.S. Rahman, and M.I. Hossain "A New Approach for LSB Based Image Steganography using SecretKey", *Proceedings of 14th International Conference on Computer and Information Technology*, IEEE Conference Publications, pp 286 – 291, 2011.

[3] S. Roy and R. Parekh, "A Secure Keyless Image Steganography Approach for Lossless RGB Images." *Proceedings of International Conference on Communication, Computing & Security*, ACM Publications, 573-576, 2011.

[4] M. Owens, *A discussion of covert channels and steganography*, SANS Institute, 2002.

[5] X. Qing., X. Jianquan and X. Yunhua., "A High Capacity Information Hiding Algorithm in Color Image.", *Proceedings of 2nd International Conference on E-Business and Information System Security*, IEEE Conference Publications, pp 1-4, 2010.

[6] R.J. Chen, Y. C. Peng, J. J. Lin, J. L. Lai, S. J. Horng, and S. J. Novel, "Multi-bit Bitwise Adaptive Embedding Algorithms with Minimum Error for Data Hiding", *Proceedings of 2010 Fourth International Conference on Network and System Security (NSS 2010)*, Melbourne, Australia, IEEE Conference Publications, 306 – 311, 2010.