A Weighted Location Based LSB Image Steganography Technique

Amitava Nag[1], Jyoti Prakash Singh[1], Srabani Khan[1], Saswati Ghosh[1], Sushanta Biswas[2], D. Sarkar[2], and Partha Pratim Sarkar[2]

[1] Department of Information Technology
Academy of Technology, West Bengal, India
[2] Department of Engineering and Technological Studies
University of Kalyani, West Bengal, India

Abstract. Steganography is the art of hiding the presence of communication by embedding secret messages into innocent, innocuous looking cover documents, such as digital images, videos, sound files. We present here a novel steganographic method based on affine cipher encryption algorithm and the least significant bit (LSB) substitution in order to provide a strong security and imperceptible visual quality to secret message. We encrypt the 8 bit secret image by changing pixel values using affine cipher. After that each 8 bit pixel of encrypted secret image is divided into 4 groups of 2 bit each. Each part which have a decimal value between 0 to 3 determines the location in each pixel of cover image where to embed the message. We do not store the actual secret message instead we encode the secret message into cover image using the value of each group of secret message. Since, we have two layers of encoding: one using private keys of affine cipher and other for steganography, our methods proves to be more secure than others. Our experimental results also proves that the proposed method has got an acceptable image quality as supported by PSNR values.

Keywords: Steganography, Affine Cipher, LSB Technique, Information Hiding, image processing.

1 Introduction

The Internet has proved to be an excellent distribution system for the digital media because of its inexpensiveness and efficiency. Due to widespread use of Internet, the sharing and transmission of images in digital form has become quite easy. However, the transmitted data can be very easily copied or modified by unauthorized persons in cyberspace. Therefore, finding ways to transmit data secretly through Internet has become an important issue. Encryption is a one of the ancient way to provide a safe way by transforming data into a cipher text via cipher algorithms [10]. Encryption techniques scrambles the message so that it cannot be understood by unauthorized users. However, this can naturally raise the curiosity level of an eavesdropper. It would be rather more prudence if

A. Abraham et al. (Eds.): ACC 2011, Part II, CCIS 191, pp. 620–627, 2011.

the secret message is cleverly embedded in another media such that the secret message is concealed to everyone. This idea forms the basis for steganography [4], which is a branch of information hiding by camouflaging secret information within other information. The word steganography in Greek means "covered writing" [6]. Steganography is the art of hiding the presence of communication by embedding secret messages into innocent, innocuous looking cover documents, such as digital images, videos, sound files [6]. In context of steganography, a message represents the information that can be embedded into a bit stream. The cover medium is an image, video, or audio signal that conceals the message. The stego-medium is the result of embedding the message in cover-medium. A possible formula of the steganographic process may be represented as

$$Cover\_medium + Embedded\_message + Stego\_key = Stego\_medium \quad (1)$$

Images provide excellent carriers for hidden information. Many different techniques have been introduced to embed messages in images [6]. The most common approaches for message hiding in images are Least Significant Bit (LSB) modification, frequency domain techniques [1,2] and spread spectrum techniques. A recent survey of these techniques is given in [6]. One of the main objective of steganography is to hide a secret message inside cover media in such a way that the secret message is not visible to the observer. The unwanted parties should not be able to distinguish between the cover-image and stego-image to prevent any sense of message inside cover. Thus the stego-image should not deviate much from original cover-image. In this article, we have proposed a novel steganographic procedure based on affine cipher encryption and location based modified Least Significant Bit (LSB) replacement. Instead of substituting the exact message in the cover image, we encode the secret message based on their weight and put on and off some bits of pixel of the cover image. We also encrypt the message prior to embedding so that even some one senses the message in cover and decodes it from cover, he or she will get a encrypted version of the message. The decryption can be done by the parties which hold the correct keys. The rest of the article is organized as follows. In section 2, we briefly discuss the related work done in the area of least significant bit (LSB) substitution steganography. We present our proposed algorithm of encryption and steganography in section 3. Section 4 presents our the experimental results and security analysis. We conclude the paper in Section 5 pointing to some future directions.

2 Related Works

By far the most popular and frequently used steganographic method is the Least Significant Bit embedding (LSB). It works by embedding message bits in the LSBs of sequentially or randomly selected pixels. The selection of pixels depends upon the secret stego key shared by the communicating parties. The popularity of the LSB embedding is due to its simplicity. The least significant bit (LSB) substitution embeds secret data by replacing k LSBs of a pixel with k secret bits directly [3]. Mathematically, the pixel value $c_{i,j}$ of the chosen pixel of cover

image for storing the k-bit message $m_{i,j}$ is modified to form the stego-pixel $s_{i,j}$ as follows:

$$s_{i,j} = c_{i,j} - c_{i,j}\%2^k + m_{i,j} \tag{2}$$

where % represent modulus operation. Many optimized LSB methods have been proposed to improve this work [12,4,8]. The human perceptibility has a property that it is sensitive to some changes in the pixels of the smooth areas, while it is not sensitive to changes in the edge areas. Not all pixels in a cover image can tolerate equal amount of changes without causing noticeable distortion. Hence, to improve the quality of stego images, several adaptive methods have been proposed in which the amount of bits to be embedded in each pixel is variable [14,13,5,9]. In 2003, Wu and Tsai proposed a novel steganographic method that uses the difference value between two neighboring pixels to determine how many secret bits should be embedded [13]. Chang and Tseng proposed a side match approach to embed secret data, where the number of bits to be embedded in a pixel is decided by the difference between the pixel and its upper and left side pixels [5]. In 2005, Wu et al. presented a novel steganographic method, which combined pixel-value differencing and LSB substitution [14]. Park et al. proposed a new method based on the difference value between two pixels adjacent to the target pixel [9]. In 2008, Wang et al. presented a steganographic method that utilizes the remainder of two consecutive pixels to record the information of secret data [11]. Yang et al. proposed an adaptive LSB steganographic method using the difference value of two consecutive pixels to distinguish between edge areas and smooth areas [15]. All pixels are embedded by the k-bit modified LSB substitution method, where k is decided by the range which the difference value belongs to [15]. Liao et. al. [7] proposed a steganographic method based on four-pixel differencing and modified least significant bit (LSB) substitution to improve the embedding capacity. A Nag et al. used transform domain technique along with Huffman coding for image steganography. In [1], they used discrete cosine transform and in [2], discrete wavelet transform to achieve quite better results in terms of security and visual quality.

3 Proposed Steganography Algorithm

Through out the article, the following notations are used.

- C represents a cover image with $c_{i,j}$ representing the value at location (i,j) of that image.
- B represents a block of the cover image with $b_{l,k}$ representing the block number (l,k)
- M represent the message with $m_{i,j}$ representing the value at location (i,j) of that message.
- E represent the encrypted message with $e_{i,j}$ representing the value at location (i,j) of that encrypted message.
- S represent the stego image with $s_{i,j}$ representing the value at location (i,j) of that stego image.

- ◇ E' represent the encrypted recovered message with $e'_{i,j}$ representing the value at location (i,j) of that message.
- ◇ M' represent the recovered message with $m'_{i,j}$ representing the value at location (i,j) of that message.
- ◇ B' represents a block of the stego image with $b'_{l,k}$ representing the block number (l,k)

Our steganography procedure consists of two phases. In first phase, we encrypt the message using affine cipher encryption method. The affine cipher encryption process changes the pixel value of location (i,j) according to the following equation

$$e_{i,j} = ((m_{i,j} \times K_1) + K_2)\%256; \tag{3}$$

where $e_{i,j}$ and $m_{i,j}$ represent encrypted value and original pixel value respectively of secret message, K_1 and K_2 are two private keys and % represent modulus operation. We replace the 4 least significant bits of the cover image with 0 by performing the operation.

$$c_{i,j} = c_{i,j} - c_{i,j}\%2^4 \tag{4}$$

where $c_{i,j}$ represents the $(i,j)^{th}$ pixel of the cover image and % represent modulus operation. The encrypted secret message is then divided into 4 groups of 2 bit each. The decimal value of each group decides the location where to put the message in the cover image. To embed the message, we also divide the cover image into blocks of 4 pixels. Depending on the value of each group of pixel in encrypted secret message, we put a 1 in each pixel of the block of the cover image. The mapping of 4 groups of secret message to 4 blocks of cover image is given in detail in algorithm 1.

3.1 The Embedding Algorithm

Algorithm 1:The embedding algorithm
Input: A gray-level cover image C of size $h \times w$, a 8 bit gray-level secret image M of size $\frac{h}{2} \times \frac{w}{2}$ and two private keys K_1 and K_2.
Output: An stego image of size $h \times w$
Steps

1. for each pixel $c_{i,j}$ of cover image C
   (a) perform $c_{i,j} = c_{i,j} - c_{i,j}\%2^4$
2. for each pixel $m_{i,j}$ of cover image M
   (a) perform $e_{i,j} = ((m_{i,j} \times K_1) + K_2)\%256$
3. Decompose C into $\frac{h}{2} \times \frac{w}{2}$ number of 2×2 blocks
4. For each pixel $e_{i,j}$ of E and block $B_{i,j}$ of C do
   (a) Divide the pixel value $e_{i,j}$ as follows
   $(b_7b_6)_3(b_5b_4)_2(b_3b_2)_1(b_1b_0)_0$.
   (b) place a 1 at $(b_7b_6)_3^{th}$ location in pixel (1,1) of $B_{i,j}$ block
   (c) place a 1 at $(b_5b_4)_2^{th}$ location in pixel (1,2) of $B_{i,j}$ block

(d) place a 1 at $(b_3b_2)_1^{th}$ location in pixel (2,1) of $B_{i,j}$ block
(e) place a 1 at $(b_1b_0)_0^{th}$ location in pixel (2,2) of $B_{i,j}$ block
5. END.

The extraction algorithm works in the reverse way by first making the recovered secret image and then decrypting the recovered secret image to get the actual secret image. The extraction process divides the stego image into blocks of 2×2. All 4 pixels of a block is read to get a pixel for recovered secret image. The complete process of recovering the secret image is given in algorithm 2. The recovered secret image is then decrypted by the following equation

$$m'_{i,j} = ((e'_{i,j} - K_2)/K_1)\%256 \tag{5}$$

where % represent modulus operation and K_1 and K_2 are the same private keys which were used during encryption.

3.2 The Extracting Algorithm

Algorithm 2:The extracting algorithm
Input: An stego image S of size $h \times w$ and two private keys K_1 and K_2 which were used for encryption
Output: The recovered image M' of size $\frac{h}{2} \times \frac{w}{2}$
Steps

1. Decompose stego image S into $\frac{h}{2} \times \frac{w}{2}$ number of 2×2 blocks
2. For each block $B'_{i,j}$ of the stego image S do
   (a) Read the 4 LSB of each pixel of block $B'_{i,j}$
   (b) For 1 in location L_i of pixel (1,1), write (b_7b_6)= which is binary equivalent of location L_i.
   (c) For 1 in location L_i of pixel (1,2), write (b_5b_4) which is binary equivalent of location L_i.
   (d) For 1 in location L_i of pixel (2,1), write (b_3b_2) which is binary equivalent of location L_i.
   (e) For 1 in location L_i of pixel (2,2), write (b_1b_0) which is binary equivalent of location L_i.
   (f) write $(b_7b_6)(b_5b_4)(b_3b_2)(b_1b_0)$ into pixel $e'_{i,j}$ of recovered encrypted secret image E'.
3. for each pixel $e'_{i,j}$ of recovered encrypted image E'
   (a) perform $m'_{i,j} = ((e'_{i,j} - K_2)/K_1)\%256$
4. END

Let us consider that one pixel of the encrypted secret image is 10010011 with a block of the cover image with 4-least significant bits replaced by 0's is given in Table 1. The bits of the secret image is divided into 4 groups as 10, 01, 00 and 11. For bit group 10, the decimal value is 2. We put a 1 in 2^{nd} position of pixel (1,1) of given block. Similarly, for group 2, 3 and 4, we put 1 in 1^{st} position of pixel (1,2), 1 in 0^{th} position of pixel (2,1) and 1 in 3^{rd} position of pixel (2,2)

Table 1. A block of the cover image with 4-LSB replaced by 0's

10100000	**11100000**
00110000	**10110000**

Table 2. A block of the cover image after insertion of secret image

1010<u>0</u>100	**1110001<u>0</u>**
0011000<u>1</u>	**1011<u>1</u>000**

respectively. The block after the changed pixels values are shown in Table 2 with changed bit are underlined.

During extraction, 1 in 2^{nd} location of pixel (1,1) gives 10, 1 in 1^{st} location of pixel (1,2) gives 01, 1 in 0^{th} location of pixel (2,2) gives 00 and 1 in 3^{rd} location of pixel (2,2) gives 11. When these values are put in order, they form the bit patterns 10010011 of recovered image.

4 Experimental Results

In this section, the performance of the proposed copyright protection scheme is evaluated and discussed. We have selected 256×256 sized 8-bit gray level image of Lena as the host image as shown in Fig. 3. The secret image is a visually recognizable 8-bit gray level image of "Ship" of size 64×64 is shown in Fig. 1. The encrypted secret generated by applying the affine cipher encryption is shown in Fig. 2. The stego image generated by embedding the encrypted secret image shown in Fig. 4. We have used the PSNR values to measure how close our stego images are to the original cover images.

Fig. 1. The secret image

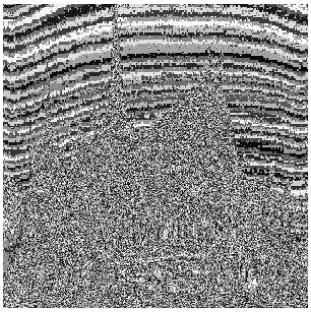

Fig. 2. The affine transformed secret image

The larger PSNR indicates that the difference between the cover-image and the stego-image is very small and this is what is desirable by any steganographic algorithm. We have got PSNR value of around 30 for all test images. The PSNR values for different images are shown in Table 3. The PSNR values as well as the visual appearance of the stego image shown in Fig. 4 suggests that the distortion level in our stego image is very less and insensitivity to human eye. Our method

Fig. 3. The cover image: Lena

Fig. 4. The Stego image: Lena

Table 3. Capacity and PSNR for different Images

Images	Size (pixel)	Capacity (pixel)	PSNR
Lena	256	64	30.48
Baboon	256	64	30.28
Airplane	256	64	30.91
Boat	256	64	30.36

does not only improves the visual quality, but also provides strong security to the message. We do not store the actual data from the secret message which makes it robust to steganalysis. Even if attackers are able to know that LSB techniques are used, normal LSB steganalysis will not suffice to get the secret image from cover image. If by some improved steganalysis, some attackers recovers the secret image. They will get an encrypted message which will be meaningless to them. They needs to find out the keys of the Affine ciphers to decrypt that message and get the actual content.

5 Conclusion

In this paper, we have proposed a novel steganographic method based on Affine cipher and location based LSB substitution. Secret data are encoded into each pixel of cover image by 4-bit LSB modification method. We do not embed the actual data instead we change the bit values of certain position in LSB of the cover image. Along with this encoding, we also perform encryption using affine cipher to encrypt the message which makes it more secure than existing steganographic techniques. Experiments show that the stego-image of our method are almost identical to the cover image. The stego image generated by our method has got just one 1 in 4 LSB of the stego image which can be a point of attack by steganalyzers. The authors are currently engaged into finding ways to mitigate this limitation and make a more robust signature free stego image.

References

1. Nag, A., Biswas, S., Sarkar, D., Sarkar, P.P.: A novel technique for image steganography based on block-dct and huffman encoding. International Journal of Computer Science and Information Technology 2(3), 103–111 (2010)
2. Nag, A., Biswas, S., Sarkar, D., Sarkar, P.P.: A novel technique for image steganography based on dwt and huffman encoding. International Journal of Computer Science and Security 4(5), 561–570 (2010)
3. Bender, D.W., Gruhl, N.M., Lu, A.: Techniques for data hiding. IBM Systems Journal 35, 313–316 (1996)
4. Chan, C.K., Cheng, L.M.: Hiding data in images by simple lsb substitution. Pattern Recognition 37(3), 469–474 (2004)
5. Chang, C.C., Tseng, H.W.: A steganographic method for digital images using side match. Pattern Recognition Letters 25(12), 1431–1437 (2004)
6. Cheddad, A., Condell, J., Curran, K., McKevitt, P.: Digital image steganography: Survey and analysis of current methods. Signal Processing 90, 727–752 (2010)
7. Liao, X., Wen, Q.-Y., Zhang, J.: A steganographic method for digital images with four-pixel differencing and modified lsb substitution. Journal Visual Communication and Image Representation 22, 1–8 (2011)
8. Lin, I.-C., Lin, Y.-B., Wang, C.-M.: Hiding data in spatial domain images with distortion tolerance. Comput. Stand. Interfaces 31, 458–464 (2009)
9. Park, Y.-R., Kang, H.-H., Shin, S.-U., Kwon, K.-R.: A steganographic scheme in digital images using information of neighboring pixels. In: Wang, L., Chen, K., S. Ong, Y. (eds.) ICNC 2005. LNCS, vol. 3612, pp. 962–967. Springer, Heidelberg (2005)
10. Stallings, W.: Cryptography and Network Security: Principles and Practices, 4th edn. Pearson Education Pvt. Ltd, India (2004)
11. Wang, C.-M., Wu, N.-I., Tsai, C.-S., Hwang, M.-S.: A high quality steganographic method with pixel-value differencing and modulus function. Journal of System Software 81, 150–158 (2008)
12. Wang, R.Z., Lin, C.F., Lin, J.C.: Image hiding by optimal lsb substitution and genetic algorithm. Pattern Recognition 34(3), 671–683 (2001)
13. Wu, D.C., Tsai, W.H.: A steganographic method for images by pixel-value differencing. Pattern Recognition Letters 24(9-10), 1613–1626 (2003)
14. Wu, H.C., Wu, N.I., Tsai, C.S., Hwang, M.S.: Image steganographic scheme based on pixel-value differencing and lsb replacement methods. Images Signal Processing 152(5), 611–615 (2005)
15. Yang, C.-H., Weng, C.-Y., Wang, S.-J., Sun, H.-M.: Adaptive data hiding in edge areas of images with spatial lsb domain systems. IEEE Transactions on Information Forensics and Security 3(3), 488–497 (2008)