An Image Steganography Technique using X-Box Mapping

Amitava Nag[1], Saswati Ghosh[2], Sushanta Biswas, Debasree Sarkar, Partha Pratim Sarkar

Academy of Technology, West Bengal University of Technology, Hoogly – 712121, India.
e-mail:[1]amitava.nag@ieee.org
[3]saswatihwh@gmail.com
biswas.su@gmail.com
dsarkar70@gmail.com
ppsarkar@klyuniv.ac.in
Dept. of Engineering and Technological Studies
University of Kalyani,
Kalyani, Nadia – 741 235, West Bengal, India

Abstract: Image steganography is a method of concealing information into a cover image to hide it. Least Significant-Bit (LSB) based approach is most popular steganographic techniques in spatial domain due to its simplicity and hiding capacity. This paper presents a novel technique for Image steganography based on LSB using X-box mapping where we have used several X-boxes having unique data. The embedding part is done by this Steganography algorithm where we use four unique X-boxes with sixteen different values (represented by 4-bits) and each value is mapped to the four LSBs of the cover image. This mapping provides sufficient security to the payload because without knowing the mapping rules no one can extract the secret data (payload).

Keywords: Steganography, X-Box, LSB Technique, Information Hiding.

1. INTRODUCTION

As the development of Internet technologies increases, the transmission of digital media is now-a-days convenient over the networks. But secret message transmissions over the Internet system suffer from serious security overhead. So, protecting of secret messages during transmission becomes an important issue. Though cryptography changes the message so that it cannot be understood but this can generates curiosity level of a hacker. It would be rather more sensible if the secret message is cleverly embedded in another media so that no one can guess if anything is hidden there or not. This idea results in steganography, which is a branch of information hiding by camouflaging secret information within other information. The word steganography in Greek means "covered writing" (Greek words "*stegos*" meaning "cover" and "*grafia*" meaning "writing") [1]. The main objective of steganography is to hide a secret message inside harmless cover media in such a way that the secret message is not visible to the observer. Thus the stego-image should not diverge much from original cover-image. In this generation, steganography is mostly used on computers with digital data being the carriers and networks being the high speed delivery channels. Figure. 1 shows the block diagram of a simple image steganographic system.

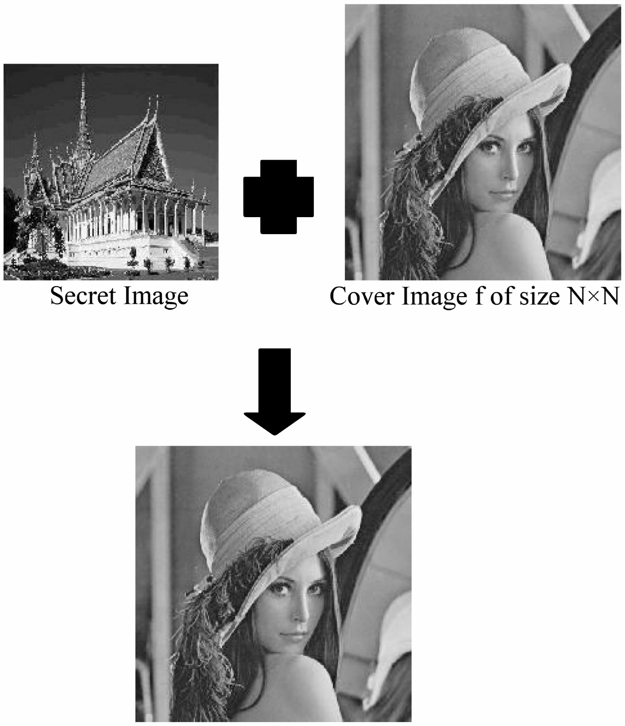


Figure. 1 The block diagram of a simple steganographic system

2 RELATED WORKS

Least significant bit (LSB) steganography [2,3,4,5,6] is the common and simple approach to embed information in a cover file. It reserves the image quality and requires no complex operation. It embeds bits of a payload into the LSB plane of a cover image. LSB matching (LSBM), LSBM revised (LSBMR) [4] and Edge Adaptive based LSBMR [5] steganography techniques are popular LSB like steganography methods.

Capacity, security and robustness [5], are the three main aspects affecting steganography and its usefulness. Capacity refers to the amount of data bits that can be hidden in the cover medium. Security relates to the ability of an eavesdropper to figure the hidden information easily. Robustness is concerned about the resist possibility of modifying or destroying the unseen data.

2.2 PSNR (Peak Signal to Noise Ratio)

The measurement of the quality between the cover image f and stego-image g of sizes N × N shown in figure 1 is defined as:

$$PSNR = 10\times\log(255^2/MSE)$$

$$\text{where } MSE = \sum_{x=0}^{N-1}\sum_{y=0}^{N-1}(f(x,y) \quad g(x,y))^2 / N^2$$

Where f(x,y) and g(x,y) means the pixel value at the at position (x, y) in the cover-image and the corresponding stego-image respectively. The PSNR is expressed in dB. The larger PSNR indicates the higher the image quality i.e. there is only little difference between the cover-image and the stego-image. On the other hand, a smaller PSNR means there is huge distortion between the cover-image and the stego-image.

3. PROPOSED IMAGE STEGANOGRAPHY ALGORITHM

Our proposed steganography technique is based on mapping the different values from X-boxes

Image Encoding:

3.1 Generation of four different X(X-OR)-boxes

X-Boxes are a 2×2 matrix, where 16 (0 to 15) values are stored as given below.

:

X-1

	0	1
0	0	4
1	13	9

X-2

	0	1
0	15	11
1	2	6

X-3

	0	1
0	10	14
1	8	3

X-4

	0	1
0	5	1
1	7	12

Figure 1.1: X-Mapping Boxes

To put values in X-boxes, we use X-OR property:
0 XOR 0 = 0 , 1 XOR 1=0 and 0 XOR 1=1 , 1 XOR 0=1.

For example 13 is inserted in any one of the four X-Boxes as follow:
13=1101=11 XOR 01=10
Thus the position of 13 is 2nd row and 1st column.

3.2 Bit Division:

Then, we need to take the cipher encrypted image; say with dimension 64×64. Now, we convert the values from decimal to binary.
For example,
The first pixel value of the encrypted image=149
Then, binary of $(149)_{10}=(10010101)_2$

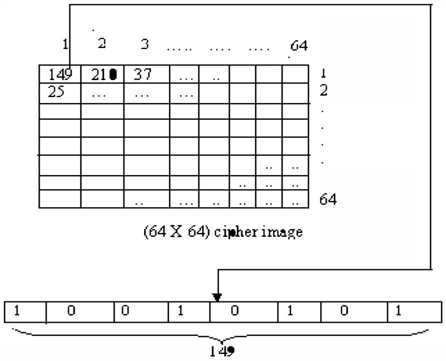


Figure 1.2: Bit Division

Now, we need to divide this 8bit values into 4parts taking 2bits in each.

$(149)_{10}=(10010101)_2$

$$\underbrace{1\ 0}_{b1}\ \underbrace{0\ 1}_{b2}\ \underbrace{0\ 1}_{b3}\ \underbrace{0\ 1}_{b4}$$

3.3 X-box Mapping:

Now we just map the values of b_1 , b_2 , b_3 , b_4 from the X-mapping box.

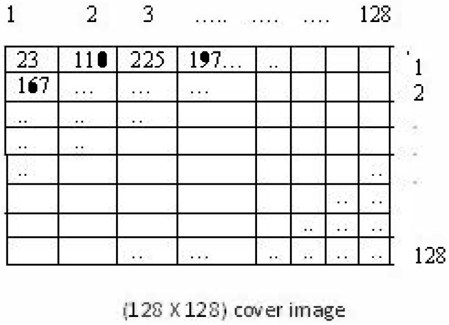
First we take b_1 =10;
Then we search the value of 1st row and 0th column of the X-1 box;

After mapping we get the value $(13)_{10}=(1101)_2$

Similarly we get mapping values for the b_2 , b_3 , b_4;
We get in the same way 11,14,1 sequentially.

3.4 Bit insertion into the cover image:

After getting the new mapping values we insert these values into the cover image. We placed these values into the 4 bit LSB of cover image sequentially. First we take the pixels one by one from the cover image. The 4 LSB bits are replaced by 13,11,14,1 sequentially.



1	2	3	…..	….	….		128	
23	110	225	197…	..				1
167	…	…	…					2
..	..	..						.
..	..							.
..							..	.
						..	..	
					..	..	..	
		..	…	..	..	..	..	128

(128 X 128) cover image

Figure 1.3

Here we take the pixels sequentially.

$(23)_{10}=(00010111)_2$;

$(110)_{10}=(01101110)_2$;

$(225)_{10}=(11100001)_2$;

$(197)_{10}=(11000101)_2$

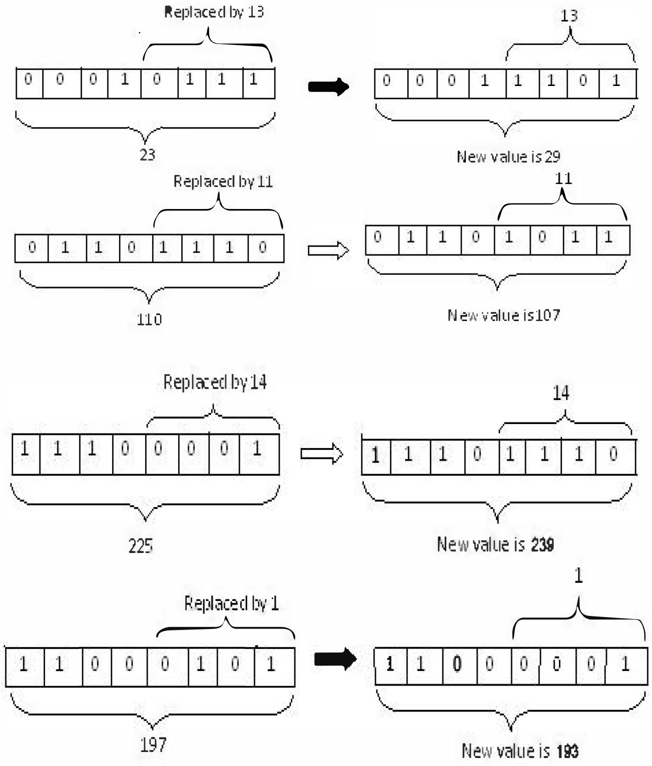
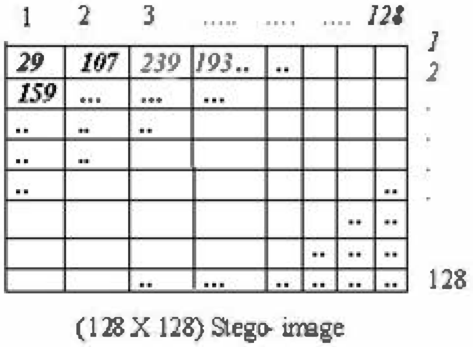


Figure 1.4 Bit Insertion into Cover Image

3.5 Formation of Stego image:

After getting the new pixel values we form the stego image. The pixel values 29, 107, 239, 193 are placed into the position of the previous values. Similarly we take the pixels one by one and insert the cipher image into them and replaced them. Thus we get the Stego-image.



1	2	3	…..	….	….		128	
29	107	239	193..	..				1
159	…	…	…					2
..	..	..						.
..	..							.
..							..	.
						..	..	
					..	..	..	
	..	…		..	..	..	..	128

(128 X 128) Stego image

Figure 1.5

These Stego image content the cipher image but we cannot recognize the cipher image. The changes of the pixel values will be varied from 0 to 15 which is a negligible amount of pixel value. So the pixel values or colors will not be change in large amount.

Encoding Algorithm

Input: A grey-level Cipher image of size (m×n), A grey-level Cover Image of size (2m×2n);
Output: Stego Image of size (2m×2n);

Steps:

1. Divide the each pixel of the cipher image into 4 parts containing 2 bits.
2. Map these 4 parts into the 4 X-boxes and get the new values for each part.
3. Insert these values into the LSB position of the Cover image one by one.
4. end

Image Decoding:

To decode the stego image in the receiver side we just perform the following steps:

3.6 Generate the 4LSB bit s from the Stego image:

We take the pixels one by one from the stego image. Transfer it into the binary values and get the 4 bits (LSB) values from it.

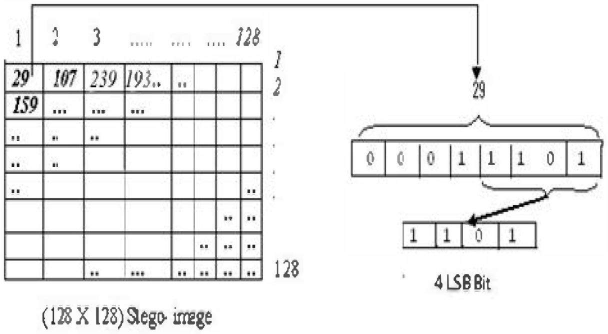


Figure 1.6: LSB (4 bits) Extraction of Stego-Image

Similarly we take the other three pixels. That is 107, 110, and 97;

$(29)_{10}=(00011101)_2$;

$(107)_{10}=(01101011)_2$;

$(239)_{10}=(11101110)_2$;

$(193)_{10}=(11000001)_2$;

LSB1=1101; LSB2=1011; LSB3=1110; LSB4=0001;

3.7 Retrieve the inserted bits of cipher image:

We take the 4 LSB bit of the stego image that are 1101, 1011, 1110, 0001; then we perform the XOR operation of the 4 bits. First we the 2 bits, and we do the XOR operation with the other 2 bits.

Lsb1=1101= 11 $\oplus$ 01= 10

Lsb2=1011= 10 $\oplus$ 11= 01

Lsb3=1110= 11 $\oplus$ 10= 01

Lsb4=0001= 00 $\oplus$ 01= 01

3.8 Concatenation of the result of the XOR operation:

Now we concatenate the 4 results of the XOR operation. After that we get the 8 bits. Then from them we transfer it into the decimal value.
Concatenated value is:

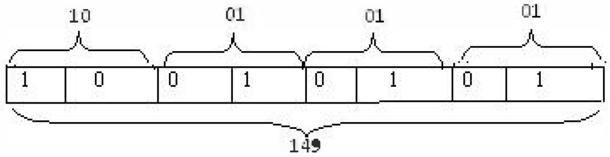


Figure 1.7: Concatination the results of XOR operation

3.9 Generation of cipher image:

Now the generated value is placed into the first position. Similarly we take the next value of the stego-image and repeat the steps 1 to 4. And we get the 210, 37 etc. Ultimately we get the total cipher image.

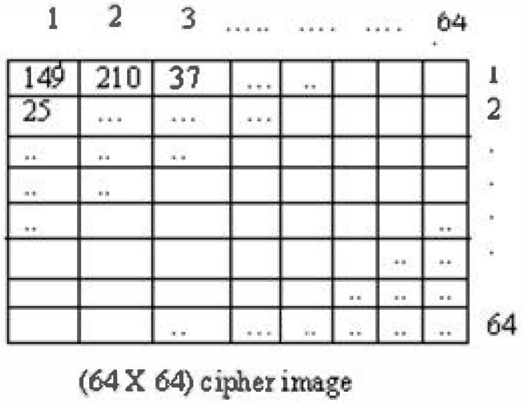


Figure 1.8: (64×64) Cipher Image

These are the total process of the X-box Steganography. Now let's see the algorithm of that particular method.

Decoding Algorithm

Input: Stego Image of size (2m×2n);
Output: A grey-level Cipher image of size (m×n);
Steps:

1. Select each pixel of the Stego-image and take 4 bits from LSB position.
2. Perform the XOR operation of that 4 bit LSB and concatenate the four results.
3. Ultimately we get the pixel value of the cipher image and place one by one to get a cipher image.
4. end

4. EXPERIMENTAL RESULTS AND SECURITY ANALYSIS

We will discuss the experimental results along with the security analysis.

4.1 Experimental Results

This embedding technique is no doubt a strongest Steganography technique than normal LSB encoding technique. Because, we embed each 2 bits of Cipher Image into the 4 bit of Cover Image. Again before insertion we coded these two bits by some mapping box into another form. So if one can understand that something is embedded in it, but the mapping will be totally unknown to him. So to extract the image is really a tough job.

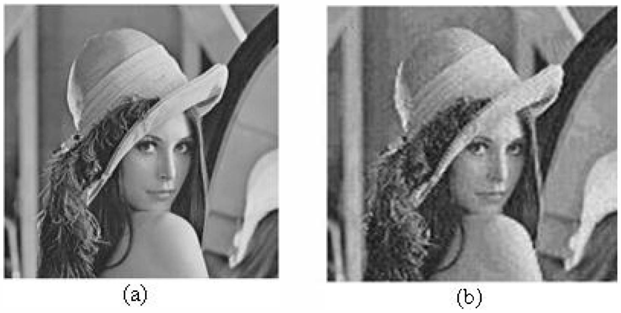


Figure 1.9: (a) Cover Image and (b) Stego Image of Lena of X-mapping box.

As we see here in the Stego Image there is no such a broad distortion. Seeing this image no one can recognize that some secret image is embedded in it. We can say that just seeing its PSNR table given below.

IMAGE NAME	Size (Pixel)	CAPACITY (%)	PSNR in (dB)
Lena.jpg	64	25%	+34.17
Baboon.jpg	64	25%	+33.98
Cameraman. jpg	64	25%	+35.42
Plane.jpg	64	25%	+35.29

Table 1.10: Capacity and PSNR of different images

5. CONCLUSION

In this paper, we propose a mapping based steganography process to improve security and image quality compared to the existing algorithms. Our approach is better because without stego key, no one can extract the original information from the stego-image, For purposes of secret communication which is more important

6. REFERENCES

[1] Moerland, T, "Steganography and Steganalysis", *Leiden Institute of Advanced Computing Science*, www.liacs.nl/home/ tmoerl/privtech.pdf

[2] C.-C. Chang, T.D. Kieu, A reversible data hiding scheme using complementary embedding strategy, Inform. Sci. 180 (16) (2010) 3045–3058.

[3] C.-C. Chang, W.-L. Tai, C.-C. Lin, A reversible data hiding scheme based on side match vector quantization, IEEE Trans. Circ. Syst. Video Technol. 16 (10) (2006) 1301–1308.

[4] W. Luo, F. Huang, J. Huang, Edge adaptive image steganography based on lsb matching revisited, IEEE Trans. Inf. Forens. Security 5 (2) (2010) 201–214.

[5] J. Mielikainen, LSB Matching Revisited, IEEE Signal Process. Lett. 13 (5) (2006) 285–287.

[6] Chen, B. and G.W. Wornell, 2001. Quantization index modulation: A class of provably good methods for digital watermarking and information embedding.IEEE Trans. Inform. Theor., 47: 1423-1443. DOI: 10.1109/18.923725.

ISBN: 978-81-909042-2-3 ©2012 IEEE