Robust Steganography based on Matching Pixel Locations

Aminou Halidou
Department of Computer Science
University of Yaounde I,
812, Yaounde, Cameroon

Youssoufa Mohamadou
Biomedical Engineering,
Energy and modeling (BEEMo)
Lab., ISST
Universite des Montagnes
208 Bangangte, Cameroon

Georges Delort Olle O.
Department of Computer Science
University of Yaounde I,
812, Yaounde, Cameroon

ABSTRACT

Steganography consist of concealing secret information in a cover object to be sent over a public communication channel. It allows two parties to share hidden information in a way that no intruder can detect the presence of hidden information. This paper presents a novel steganography approach based on pixel location matching of the same cover image. Here the information is not directly embedded within the cover image but a sequence of 4 bits of secret data is compared to the 4 most significant bits (4MSB) of the cover image pixels. The locations of the matching pixels are taken to substitute the 2 least significant bits (2LSB) of the cover image pixels. Since the data are not directly hidden in cover image, the proposed approach is more secure and difficult to break. Intruders cannot intercept it by using common LSB techniques.

General Terms

Image processing; Security; Steganography

Keywords

Steganography, 4MSB, 2LSB, Matching Pixel Locations.

1. INTRODUCTION

Nowadays, information can be transmitted conveniently over the most popular medium of communication called the Internet. Internet communication plays a great role in bringing individuals, communities and organizations closer together through information sharing. However there is also an increasing risk of information theft and piracy[1, 2] . The transmission of secret information can be achieved through two technics: Cryptography and Steganography[3]. In cryptography, the secret data is scrambled and the information is extracted by the parties who own the secret key[4]. Steganography is one of the modern world communication techniques used to hide information (see Figure 1).

Like signal modulation, the steganography hides information inside a cover medium as carrier. Steganography is the art of secret communication which uses one data to hide another one, the existence of the secret data is covered[5]. All start with a cover object such as text, image, audio and video used to embed the secret information. Among the cited cover media, digital images are preferred due to their high frequency over the internet[6]. The image in which the secret data has to be hidden is called the cover image[7] while the image containing the secret data is refers as stego-image (see Figure 2).

Steganalysis is the algorithm used to break the steganoghraphy protection. To implement a good steganoghraphy, it is very important to make stego-image satisfies the following characteristics:

- Imperceptibility: no difference between cover image and stego-image using naked human eyes.
- Secure: unintended users should not suspect the stego-image containing secret data.
- Robustness: secret information is not altered by noise.
- Statistically undetectable: using statistical/mathematical attacks there must be little difference between cover and stego[8].
- Embedding capacity: how much data can be hidden in a single stego-image

The PSNR (Peak Signal to Noise Ratio) is used to calculate the quality of a given image. It is calculated thus; PSNR=10log$_{10}$ (255$_2$/MSE). Higher PSNR value indicates better steganography algorithm and high similarity between stego-image and cover image while Smaller PSNR value implies poor steganography algorithm and high dissimilarity between stego-image and cover image. MSE is the average Mean Square Error between the stego-image and the cover image,[9, 10].

The remaining part of this paper is organized as follows. In section II the previous approaches are discussed. Section III and IV present the proposed method and algorithms. Experiments and results are discussed in section V. In section VI the work is concluded.

2. PREVIOUS WORK

Many techniques and algorithms have been proposed in the last decade for steganography. They can generally be grouped under spatial domain steganography and transform domain steganography. In transform domain steganography pixels of the cover image is transformed in to frequency domain coefficients that used to hide the secret data. The transforms usually used in this method are Discrete Fourier Transform (DFT)[11], Discrete Cosine Transform (DCT)[12], and Discrete Wavelet Transform (DWT)[13]. In the spatial domain, LSB technique is considered as the most common, advantageous, and easiest to implement. LSB approach uses the lowest significant bit of each pixel of the image. Basically the secret bit stream is hidden within the least significant bits of pixels of the cover image. Chan, C. K., & Cheng, L. M used optimal pixel adjustment process (OPAP) to embedded data and enhance stego-image quality [14]. Any given color image consists of pixels of three bytes for color image and one byte for gray level image. The secret bits to be transmitted are embedded in these bytes in certain manner, see Table 3.

Given an example of gray image with the first eight pixels as follow: 10010010 01101010 10011111 10101100 01010101 01000111 00110110 01001011. Using h whose binary value is 01101000 as the secret data. The replacement will be as follows: 1001001**0** 0110101**1** 1001111**1** 1010110**0** 0101010**1** 0100011**0** 0011011**0** 0100101**0**. The change of the LSB's of the pixel values is difficult to observe by naked eyes. Since the gray levels (colors) of the stego-image is very similar to those of the cover image. However, using the simplest LSB make the system extremely vulnerable to attacks.

In another work, R. Chandramouli and N. Memon studied the steganography capacity and they found the upper bound of this capacity, i.e how many bits can be substituted without statically modifying the image. They used a method in which the cover image pixel are selected in pseudo-random order generated by a secret key, if there is a match between the LSB of cover pixel and the secret stream bits then no change otherwise and addition or subtraction is perform[15]. Ching-Sheng Hsu and Shu-Fen Tu proposed Ant Colony optimization algorithms to find the optimal LSB substitution. The optimal point is found using an optimal matrix[16]. Da-Chun Wua, Wen-Hsiang Tsaib used the difference of two pixels of color cover image to embed secret bits. The cover image is divided into non-overlapping regions of two consecutive pixels[17]. Prince Kumar Panjabi proposed an approach for hiding data within an image using four modules – minimum differencing function, mapping rules, pixel selection method, and set classifier method. Secret message is mapped using the Minimum Pixel Difference function[18]. Pixel intensity of dark region was also considered for data hiding[19]. To enhance the robustness of their method, Karthikeyan et al. adjusted the intensity of the cover image, and hid the secret data in two non- consecutive pixels LSBs[20]. W. Luo et al. proposed a simple LSB embedding technique, in which the region that is selected to hide the data is determined using edge adaptive algorithm, which takes into account the size of the secret data and the difference of two consecutive pixels of the cover image[21]. D. Rawat and V. Bhandari used the LSB of color image to embed the MSB of secret data. Since the research made by Hecht shows that 65% of all human cones are sensitive to red, 33% to green and 2% to blue, the MSB of secret data is inversely proportional distributed[22]. Kh. Manglem Singh, L Shyamsudar Singh, and Buboo Singh proposed a LSB embedding method in which the secret message is hidden in edges of the image following the non-adjacent pixel locations[23]. In this work the embedding capacity is determine by calculating the difference value of two pixels that are consecutive[24].

3. PROPOSED WORK

The purpose of this work is to hide secret data within a grey scale image in a manner that no intruder can know its existence. To do this the 4MSB, 2LSB, and location of cover image pixels are used. Two algorithms are used to implement the system:

- Embedding algorithm, for hiding the secret bits stream in the cover image.
- Extraction algorithm, for extracting the embedded bits from the stego-image.

3.1 Hiding the Secret Data

To embed the secret data, a secret text and cover-image are read and each character of the secret text converted into binary using ASCII system (8 bits for each character) and stored in and array (Array1). The 4MSB of each pixel of the cover-image is compared to a sequence of 4 bits of the binary Array1. If there is a match, the location of the pixel is saved in Array2, otherwise the cover-image is changed and the processes are repeated. In this manner each sequence of 4 bits are covered till the last 4 bits of Array1. Lastly, the 2LSB of the cover-image pixels are substituted with 2 bits of Array2 sequentially and saved as stego-image to be sent (see Figure 3).

3.2 Secret Data Extraction

To extract the secret data, the stego-image is opened and the 2LSB of each pixel is extract and saves in Array3. Array3 is used to get pixel locations which are saved in Array4. Using Array4 the 4MSB of pixels are extracted and saved in Array5, which is in turn converted to text (ASCII to text) see Figure 4.

4. ALGORITHMS

4.1 Embedding Algorithm

Input: Secret text, cover-image.
Output: Stego-image.
Step1: Read the text and cover-image as inputs.
Step2: Use ASCII system to convert the text to binary and save it in Array1.
Step3: Take a sequence of 4 bits from Array1 and compare with 4MSB of each pixel.
Step4: If comparison is matched then save the pixel location in Array2, otherwise change the cover-image and start the operation again.
Step5: Move to the next 4 bits of array1 and repeat step3 to step4 till the terminating data is found.
Step6: Take sequentially 2 bits of Array2 to substitute the 2LSB of each pixel.
Step7: Save the stego-image to be sent.
Step8: End.

4.2 Information Extraction Algorithm

Input: stego-image.
Output: Secret text.
Step1: Get the stego-image as input.
Step2: Extract the 2LSB of each pixel, concatenate and save in Array3.
Step3: Find pixel location using binary number in Array3.
Step4: For each corresponding location, extract the 4MSB of the pixel and save in Array4.
Step5: Convert binary number in Array4 to text using ASCII system and save the file.
Step6: End

5. EXPERIMENTAL RESULTS AND ANALYSIS

The experiment was done on Intel(R) Core(TM) i5-3317U CPU @ 1.70GHz, using Matlab R2014a 64-bit, the results of the proposed example is as illustrated in the following.

5.1 Information Embedding

a) Read the cover-image (gray image) and set the 4LSB to 0 (see Figure 5).
b) Read the secret text: "hello"
c) Equivalent Decimal / ASCII Value

104	101	108	108	111

d) Binary conversion of (c)

01101000	01100101	01101100	01101100	01101111

e) 4bits transformation of (d)

0110	1000	0110	0101	0110

1100	0110	1100	0110	1111

f) Zero padding of (e)

01100000	10000000	01100000	01010000	01100000
11000000	01100000	11000000	01100000	11110000

g) Decimal conversion of (f)

96	128	96	80	96	192	96	192	96	240

h) Pixels locations corresponding to (g)

106, 7	102, 8	106, 7	104, 7	106, 7	112, 9	106, 7	112, 9	106, 7	118, 10

i) Binary conversion of (h)

01101010, 00000111	01100110, 00001000	01101010, 00000111	01101000, 00000111
01101010, 00000111	01110000, 00001001	01101010, 00000111	01110000, 00001001
01101010, 00000111	01110110, 00001010		

j) Concatenated Binary value of (i)

0110101000000111011001100000010000110101000000111011010000000001110110101000000111011100000000100101101010000001110111000000001001011010100000001110111011000001010

k) Replace the 2LSB of cover image with (j) values (stego-image is built) see Figure 6.

5.2 Information Extraction

a) Read the stego-image and extract the 2LSB of pixels

0110101000000111011001100000010000110101000000111011010000000001110110101000000111011100000000100101101010000001110111000000001001011010100000001110111011000001010

b) Transform (a) to pixel locations

01101010, 00000111	01100110, 00001000	01101010, 00000111	01101000, 00000111
01101010, 00000111	01110000, 00001001	01101010, 00000111	01110000, 00001001
01101010, 00000111	01110110, 00001010		

c) Decimals conversion of (b)

106,7	102,8	106,7	104,7	106,7	112,9	106,7	112,9	106,7	118,10

d) Set the stego-image 4LSB to 0 (see Figure 7)

e) Extract pixels values from d) using c)

96	128	96	80	96	192	96	192	96	240

f) Binary conversion of e)

01100000	10000000	01100000	01010000	01100000
11000000	01100000	11000000	01100000	11110000

g) Remove 4LSB of f)

0110	1000	0110	0101	0110
1100	0110	1100	0110	1111

h) 8 Bits concatenation of g)

01101000	01100101	01101100	01101100	01101111

i) Equivalent Decimal / ASCII Value of h

104	101	108	108	111

j) Corresponding String: hello (secret text: "hello")

6. DISCUSSION

Since the final stego-image sent only has its 2LSB modified this makes the difference with the cover image imperceptible. However the fact that the stego-image only holds pixel location information this makes the algorithm more robust and secure than the simple LSB technique. Another characteristic of a good steganography algorithm is its capacity to embed secret data. In the proposed algorithm the embedding capacity is lower than that of simple LSB technique since eight (8) pixel locations in the stego-image are used to hide four (4) bits.

However, a simple but effective technique was used to increase the embedding capacity of the algorithm. The technique makes use of fixed length coding to encode the locations of the pixels before substitution. To explain this consider first a 2LSB technique. The embedding capacity of the algorithm will be 2xMxN, where MxN is the size of the cover image. In the proposed algorithm the embedding capacity is given by

$$E_C = \frac{2\mathrm{MN}}{n_{loc}} n_{MSB}$$

Where n_{loc} the number of bits used to identify a given location (in this case 16 bits) and n_{MSB} is the number of unpadded bits used to hide the data in step above (in this case 4 bits).

Now the fact that zero padding is used on the secret data and stego-image limits the possible values the data and the gray levels of the stego-image to sixteen (16) different values. This increases the chance of a data marching a given pixel value there by ensuring that all chosen cover images will have similar embedding capacity. Considering a gray level image with 256 gray levels, the possible values of gray levels after padding will be given as column 1 of table 1.

The locations of the pixels with these values are given in column 2/3. Since there were 16 locations, a fixed length code of 4 bits was used to encode these locations as shown in column 4. These bits are then used to substitute the 2LSB of the cover image. Doing so multiplied the embedding capacity by 4. So the new embedding capacity will be the same as that of a simple 2LSB technique.

Table 1. Pixel location encoding to increase embedding capacity

Gray level	Location		4 bit encoding
0	104, 6	01101000, 00000110	0000
16	101, 8	01100101, 00001000	0001
32	96, 8	01100000, 00001000	0010
48	107, 9	01101011, 00001001	0011
64	113, 1	01110001,	0100

		00000001	
80	104, 7	01101000, 00000111	0101
96	106, 7	01101010, 00000111	0110
112	103, 3	01100111, 00000011	0111
128	102, 8	01100110, 00001000	1000
144	114, 9	01110010, 00001001	1001
160	110, 2	01101110, 00000010	1010
176	116,8	01110100, 00001000	1011
192	112, 9	01110000, 00001001	1100
208	114, 7	01110010, 00000111	1101
224	110, 8	01101110, 00001000	1110
240	118,10	01110110, 00001010	1111

Additionally, Huffman coding could be used to encode the pixel location bit stream before replacing the 2LSB of the cover image pixel values[25][26]. This will increase the embedding capacity of the algorithm. Table 2 shows some benchmark parameters for the evaluation of the proposed steganography of algorithm. The evaluation was done on six different cover images. The payload capacity indicates the maximum number of characters that can be hidden on a cover image. As explained before it can be seen that the embedding capacities of all the six cover images are the same. The peak signal to noise ratio (PSNR) is the ratio between a signal‘s maximum power and the power of the signal‘s noise. The PSNR indicates the quality of the stego-image after embedding the secret message in the cover in other words how much is the resemblance of the stego-image to the cover image. High values of PSNR are better.

Table 2. PSNR Result

Cover image (512×512 Grey Level)	**Payload Capacity (Characters)**	**PSNR (in dB)**
Baboon	65532	49.73
Tiffany	65532	51.68
Landscape	65532	51.57
Lena	65532	51.81
Peppers	65532	51.71
Boat	65532	51.62

Table 1 shows the image quality comparison parameters computed between the cover images and stego-images.

Figure 8 Different Histogram shows also the histogram of cover image and stego-image.

7. CONCLUSION

Steganography is widely used to secure today’s communication world, its primary goal is to hide data in such a way that only approved users can decode and read it. The experiment has been done over a set of standard images: Baboon, Landscape, Sailboat, Tiffany, Peppers, and Lena. The proposed approach proved to be efficient to hide a text in image by using 4MSB, 2LSB, and some pixels position. Instead of hiding the text in the image it uses the corresponding pixels location, which makes it secure and easy to use. This technique can be exploited for copyright protection, sending symmetric key over Internet, and storing sensitive information in storage system such as database. The ameliorated system can be used to encode images and videos in future.

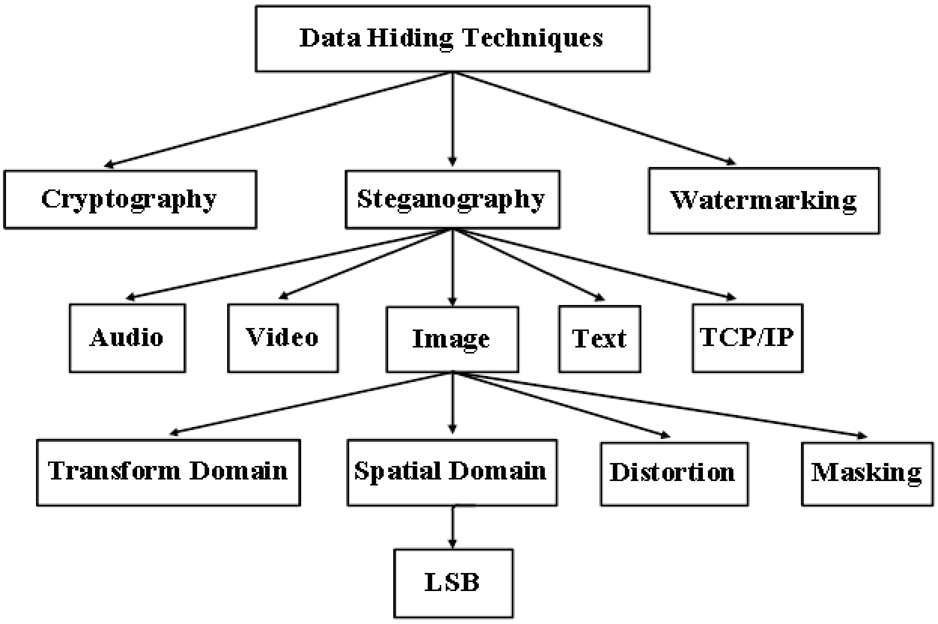

Figure 1: classification of data hiding

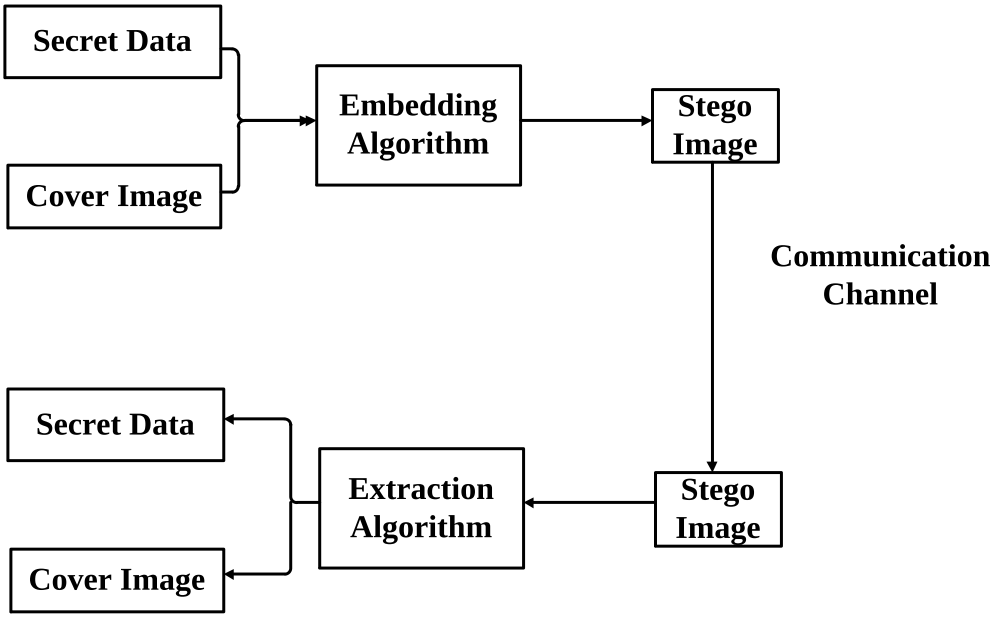

Figure 2: Steganography system

Table 3: LSB Steganography of gray level image

Cover Image Pixels								**Secret Byte**	**Stego-image Pixels**							
1	0	0	1	0	0	1	0	**0**	1	0	0	1	0	0	1	**0**
0	1	1	0	1	0	1	0	**1**	0	1	1	0	1	0	1	**1**
1	0	0	1	1	1	1	1	**1**	1	0	0	1	1	1	1	**1**
1	0	1	0	1	1	0	0	**0**	1	0	1	0	1	1	0	**0**
0	1	0	1	0	1	0	1	**1**	0	1	0	1	0	1	0	**1**
0	1	0	0	0	1	1	1	**0**	0	1	0	0	0	1	1	**0**
0	0	1	1	0	1	1	0	**0**	0	0	1	1	0	1	1	**0**
0	1	0	0	1	0	1	1	**0**	0	1	0	0	1	0	1	**0**

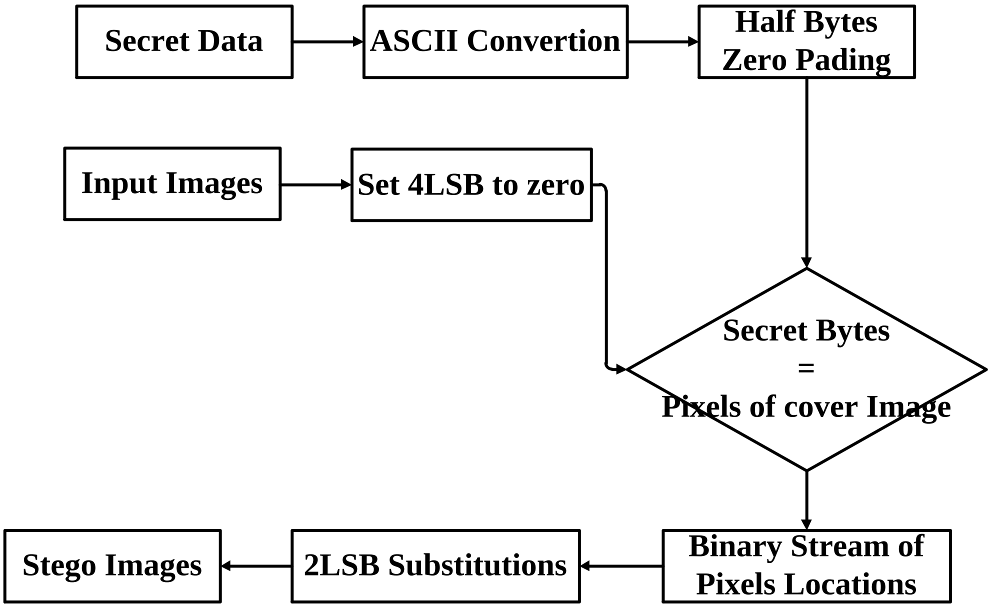

Figure 3: Hiding Secret Data

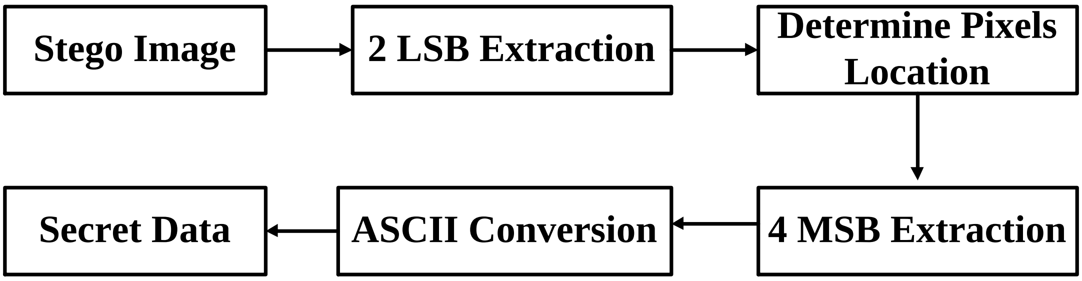

Figure 4: Extracting Secret Data

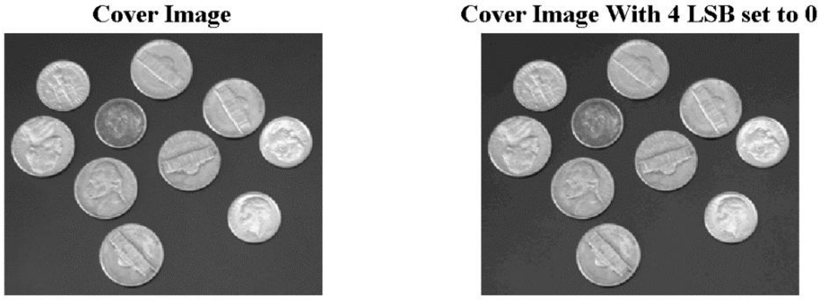

Figure 5: Cover Image with 4 LSB set to 0

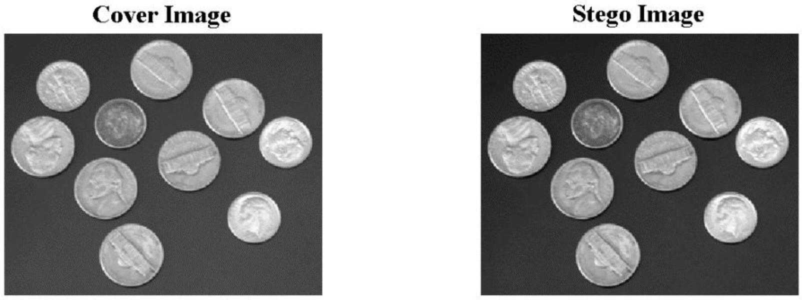

Figure 6: Cover Image & Stego-image

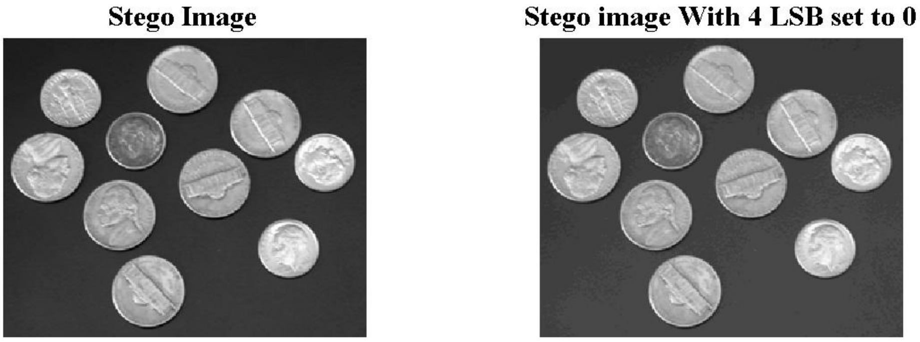

Figure 7: Stego-image with 4 LSB set to 0

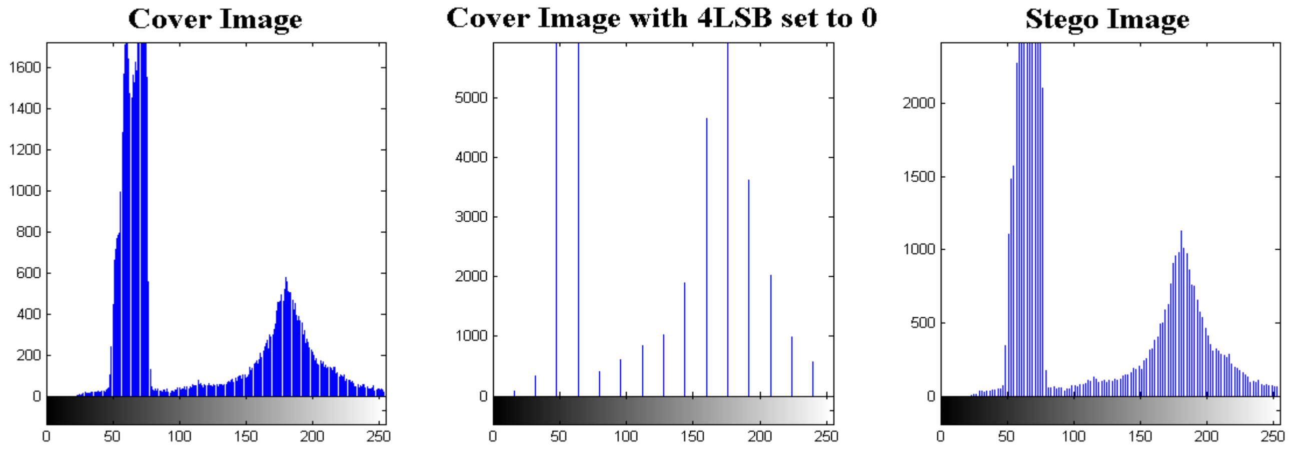

Figure 8: Different Histogram

8. REFERENCES

[1] D. Wall, Cybercrime: The transformation of crime in the information age vol. 4: Polity, 2007.

[2] T. J. Holt and M. G. Turner, "Examining risks and protective factors of on-line identity theft," Deviant Behavior, vol. 33, pp. 308-323, 2012.

[3] S. A. Laskar and K. Hemachandran, "High Capacity data hiding using LSB Steganography and Encryption," International Journal of Database Management Systems, vol. 4, p. 57, 2012.

[4] B. Schneier, Schneier's Cryptography Classics Library: Applied Cryptography: Protocols, Algorithms, and Source Code in C, 1996.

[5] L. Ji, X. Li, B. Yang, and Z. Liu, "A further study on a PVD-based steganography," in Multimedia Information Networking and Security (MINES), 2010 International Conference on, 2010, pp. 686-690.

[6] S. Bhattacharyya, A. P. Kshitij, and G. Sanyal, "A novel approach to develop a secure image based steganographic model using integer wavelet transform," in Recent Trends in Information, Telecommunication and Computing (ITC), 2010 International Conference on, 2010, pp. 173-178.

[7] C.-C. Chang, J.-Y. Hsiao, and C.-S. Chan, "Finding optimal least-significant-bit substitution in image hiding by dynamic programming strategy," Pattern Recognition, vol. 36, pp. 1583-1595, 2003.

[8] A. Martin, G. Sapiro, and G. Seroussi, "Is image steganography natural?," IEEE Transactions on Image processing, vol. 14, pp. 2040-2050, 2005.

[9] E. Ghasemi, J. Shanbehzadeh, and N. Fassihi, "High capacity image steganography based on genetic algorithm and wavelet transform," in Intelligent Control and Innovative Computing, ed: Springer, 2012, pp. 395-404.

[10] G. Swain and S. K. Lenka, "Classification of image steganography techniques in spatial domain: a study," Int. J. Comput. Sci. Eng. Tech.(IJCSET), vol. 5, pp. 219-232, 2014.

[11] C. Chen and Y. Q. Shi, "JPEG image steganalysis utilizing both intrablock and interblock correlations," in Circuits and Systems, 2008. ISCAS 2008. IEEE International Symposium on, 2008, pp. 3029-3032.

[12] K. S. Kumar, K. Raja, R. Chhotaray, and S. Pattanaik, "Bit length replacement steganography based on dct coefficients," International Journal of Engineering Science and Technology, vol. 2, pp. 3561-3570, 2010.

[13] S. Bhattacharyya and G. Sanyal, "A robust image steganography using dwt difference modulation (DWTDM)," International Journal of Computer Network and Information Security, vol. 4, p. 27, 2012.

[14] C.-K. Chan and L.-M. Cheng, "Hiding data in images by simple LSB substitution," Pattern recognition, vol. 37, pp. 469-474, 2004.

[15] R. Chandramouli and N. Memon, "Analysis of LSB based image steganography techniques," in Image Processing, 2001. Proceedings. 2001 International Conference on, 2001, pp. 1019-1022.

[16] C.-S. Hsu and S.-F. Tu, "Finding optimal LSB substitution using ant colony optimization algorithm," in Communication Software and Networks, 2010. ICCSN'10. Second International Conference on, 2010, pp. 293-297.

[17] D.-C. Wu and W.-H. Tsai, "A steganographic method for images by pixel-value differencing," Pattern Recognition Letters, vol. 24, pp. 1613-1626, 2003.

[18] P. K. Panjabi and P. Singh, "An Enhanced Data Hiding Approach Using Pixel Mapping Method with Optimal Substitution Approach," International Journal of Computer Applications, vol. 74, 2013.

[19] H. Motameni, M. Norouzi, M. Jahandar, and A. Hatami, "Labeling method in Steganography," in Proceedings of world academy of science, engineering and technology, 2007, pp. 349-354.

[20] B. Karthikeyan, V. Vaithiyanathan, B. Thamotharan, M. Gomathymeenakshi, and S. Sruti, "LSB replacement steganography in an image using pseudorandomised key generation," Research Journal of Applied Sciences, Engineering and Technology, vol. 4, pp. 491-494, 2012.

[21] W. Luo, F. Huang, and J. Huang, "Edge adaptive image steganography based on LSB matching revisited," IEEE transactions on information forensics and security, vol. 5, pp. 201-214, 2010.

[22] D. Rawat and V. Bhandari, "A steganography technique for hiding image in an image using lsb method for 24 bit color image," International Journal of Computer Applications, vol. 64, 2013.

[23] K. M. Singh, L. S. Singh, A. B. Singh, and K. S. Devi, "Hiding secret message in edges of the image," in Information and Communication Technology, 2007. ICICT'07. International Conference on, 2007, pp. 238-241.

[24] C.-H. Yang, C.-Y. Weng, S.-J. Wang, and H.-M. Sun, "Adaptive data hiding in edge areas of images with spatial LSB domain systems," IEEE Transactions on Information Forensics and Security, vol. 3, pp. 488-497, 2008.

[25] A. Nag, J. P. Singh, S. Biswas, D. Sarkar, and P. P. Sarkar, "A Huffman code based image steganography technique," in International Conference on Applied Algorithms, 2014, pp. 257-265.

[26] N. Pandian, "An Image Steganography Algorithm Using Huffman and Interpixel Difference Encoding," International Journal of Computer Science & Security (IJCSS), vol. 8, p. 202, 2014.