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

International Journal of Advanced Research in Computer and Communication Engineering
Vol. 4, Issue 6, June 2015

High Capacity Steganography Method Based upon RGBA Image

Nawar S. Alseelawi[1], Tarik Z. Ismaiel[2], Firas A. Sabir[3]

M.Sc. Student, Department of Computer Engineering, College of Engineering, Baghdad, Iraq [1]

Professor, Department of Electrical Engineering, College of Engineering, Baghdad, Iraq [2]

Professor, Department of Computer Engineering, College of Engineering, Baghdad, Iraq [3]

Abstract: One of the most important factors of information technology and communication has been the security of the information. For security purpose the concept of Steganography is being used. Imperceptibility and hiding capacity are very important aspects for efficient secret communication. In this paper a new steganography approach proposed based on LSB technique by using ALPHA channel on JPG cover images and Bit-slicing decomposition on the secrete image. for this method first the secrete image decomposed to bit streams and the data encrypted using an encryption method. On the cover side, an alpha channel is attached to the cover image and the data embedded into LSBs of RGBA channels. The method was implemented and tested by using MATLAB® (R2011a).

Keywords: LSB, ALPHA channel, RGBA, Bit-slicing.

I. INTRODUCTION

Steganography is an ancient art that has been reborn in recent years. The word Steganography comes from Greek roots which literally means "covered writing", and is usually interpreted to mean hiding information in between other information [1]. A steganography system is expected to meet three key requirements, namely transparency, capacity and robustness. Transparency evaluates the image distortion due to signal modifications like message embedding or attacking. Capacity: It is the maximum amount of information that a data hiding scheme can successfully embed without introducing any perceptual distortion in the marked media. Robustness measures the ability of embedded data or watermark to withstand against intentional and unintentional attacks [2].

Steganographic methods can be broadly classified based on the embedding domain, digital steganography techniques are classified into (i) spatial domain, (ii) frequency domain. In Spatial domain image steganography, cover image is first decomposed in to its bits planes and then LSB's (Least Significant Bits) of the bits planes are replaced with the secret data bits. As LSB's are redundant bits and contributes very less to overall appearance of the pixel, replacing it has no perceptible effect on the cover-image. Advantages are high embedding capacity, ease of implementation and imperceptibility of hidden data. The major drawback is its vulnerability to various simple statistical analysis methods.The most direct way to represent pixel's colour is by giving an ordered triple of numbers: red (R), green (G), and blue (B) that comprises that particular colour. The other way is to use a table known as palette to store the triples, and use a reference into the table for each pixel. For transparent images, extra channel called the Alpha value is stored along with the RGB channels. RGBA image stands for Red, Green, Blue, and Alpha. It extends the RGB colour model with the alpha value representing the transparency of pixels. The A value varies from 0 to 255, in which 0 means completely transparent while 255 means opaque. PNG images follow the RGBA colour model [3]. Bit-plane slicing decomposition highlighting the contribution made to the total image appearance by specific bits. Assuming that each pixel is represented by 8-bits, the image is composed of eight 1-bit planes. Plane (0) contains the least significant bit and plane (7) contains the most significant bit. Only the higher order bits (top four) contain the majority visually significant data. The other bit planes contribute the more subtle details [4].There are many researches in each of the steganography techniques, and a brief description of some of this research is presented: For the researches which are presented the high capacity steganography methods are [5-7].In this work an alpha channel is attached to a cover image with RGB colour system (24 bits depth), the resulting image is a PNG (Portable Network Graphics) image with RGBA colour system (32 bits depth), on the other hand, using Bit-plane Slicing decomposition on the secrete image to compress it and transform the gray-level secrete image to a binary bit stream, then the secrete message bit streams will encrypted with a key and embedded in the four colour planes of the cover image.

II. THE PROPOSED TECHNIQUE

While most of steganography techniques work on cover image or secrete image, our proposed technique relies on processing both of cover and secrete image to reach to the optimum results.

For the secrete image side the total data size is decreased, i.e. compressing the image to decrease the amount of the payload. Bit-plane slicing technique used to compress the secrete image and also to convert it from 2D image to 1D bit stream. On the other side, working on the cover image to increase its ability to handle the payload. A fourth channel added to the JPG cover image to increase the bit depth from 24 to 32 , and to be four channels carrying the

DOI 10.17148/IJARCCE.2015.4627

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

International Journal of Advanced Research in Computer and Communication Engineering
Vol. 4, Issue 6, June 2015

four candidate bit-planes. The proposed system is presented in the Fig. 1 for the sender side, and Fig. 2 for the receiver side.

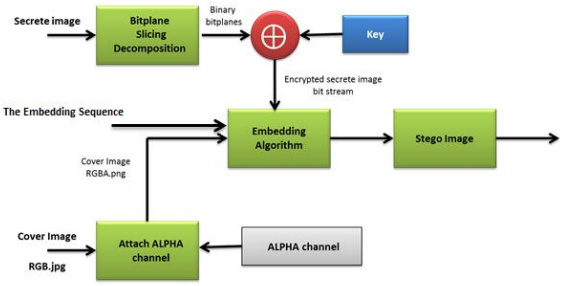


Fig. 1 The Main block diagram of the sender side

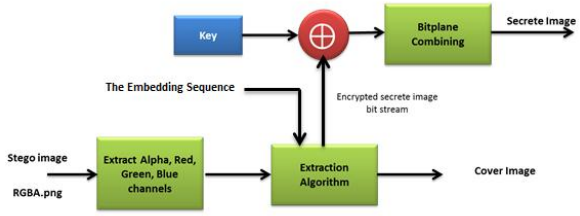


Fig.2 The Main block diagram of the reciever side

A. Preparation of the Cover Image

Given cover image is a colour image. Let A be an original colour image having size $m * n * p$ represented as:

$$A = \left\{ x(i,j,k) \middle| \begin{matrix} 0 \le i < m, 0 \le j < n \;\; 0 \le k < p \\ x(i,j,K) \epsilon \{0,1,2,3,4, \ldots \ldots, 255\} \end{matrix} \right\} \ldots (1)$$

Value of K varies from 1 to 3.

This image has the extension of JPG. It has 3 colour channels (Red, Blue, Green). To add the fourth channel, it must be defined :

$$alpha = \left\{ x(i,j) \middle| \begin{matrix} 0 \le i < m, 0 \le j < n \\ x(i,j) = 255 \end{matrix} \right\} \ldots (2)$$

The size of alpha channel is exactly same to that of cover colour image. In this work, the Alpha chose to be all ones, and this mean alpha channel will be a white plane acts as a transparent background of the image.

B. Preparation and Decomposition of Secrete Image

Let G is a secrete grayscale image having size $r * c$ represented as :

$$G = \left\{ x(i,j) \middle| \begin{matrix} 0 \le i < r, 0 \le j < c \\ x(i,j) \epsilon \{0,1,2,3,4, \ldots \ldots, 255\} \end{matrix} \right\} \ldots (3)$$

This 2D secrete grayscale image is first passed through bit-plane slicing algorithm.For grayscale image having 8 bit-planes, this can be represented as follows:

$$Pl_k = \left\{ x(i,j,k) \middle| \begin{matrix} 0 \le i < r, 0 \le j < c \\ x(i,j,k) \epsilon \{0,1\} \end{matrix} \right\} \ldots (4)$$

Where: $1 \le k \le 8$

The 8th bit-plane contain more information than other plane, then for embedding process could only choose 8th , 7th , 6th and 5th bit-plane.To convert all the 4 selected bit-planes into 1D array as shown below :

$1 * (r * c)$ and represented as follows for each of 4 upper bit-planes :

$$sec_{array} = \left\{ x(1,j) \middle| \begin{matrix} 0 \le j < (r * c) \\ x(1,j) \epsilon \{0,1\} \end{matrix} \right\} \ldots (5)$$

To combine all four strings into 1D binary secrete array :

$$sec_{total} = [sec_{array5} sec_{array6} sec_{array7} sec_{array8}] \ldots (6)$$

Secrete 1D array then divided into 4 parts. Then by finding the length of the$sec_{total}$and the length of each divided string.

Suppose this length islen :

$$a_a = len/4 \quad \ldots (7)$$

$$b_b = a_a + a_a \quad \ldots (8)$$

$$c_c = a_a + a_a + a_a \quad \ldots (9)$$

The content of the 1st secrete string is :

$$sec_{str1} = sec_{total}(1 : a_a) \ldots (10)$$

The content of the 2nd secrete string is :

$$sec_{str2} = sec_{total}(a_a + 1 : b_b) \ldots (11)$$

The content of the 3rd secrete string is :

$$sec_{str3} = sec_{total}(b_b + 1 : c_c) \quad \ldots (12)$$

And the content of the 4th secrete string is :

$$sec_{str4} = sec_{total}(c_c + 1 : end) \quad \ldots (13)$$

C. Encryption of the secrete image

For more security, encryption is applied to the secrete image. In this work, a simple encryption algorithm using a private key used to encrypt the secret image. Secrete key is a binary array with length varies for every secrete string.

$$sec\_key = \left\{ x(1,j) \middle| \begin{matrix} 0 \le j < sec_{total} \\ x(1,j) \epsilon \{0,1\} \end{matrix} \right\} \ldots (14)$$

Secrete key and secrete messages are XORed with each other's to produce encrypted secrete strings as follows :

$$enc\_sec\_img1 = \{sec_{str1} \oplus sec\_key\}$$

$$enc\_sec\_img2 = \{sec_{str2} \oplus sec\_key\} \quad \ldots (15)$$

$$enc\_sec\_img3 = \{sec_{str3} \oplus sec\_key\}$$

$$enc\_sec\_img4 = \{sec_{str4} \oplus sec\_key\}$$

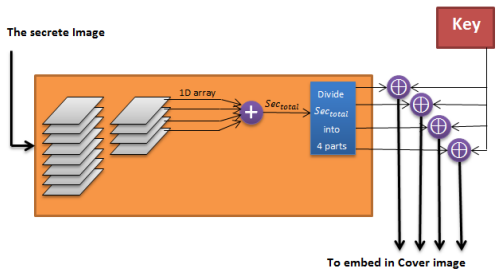


Fig.3 Encryption of Secrete image using Simple Private Key

D. The Embedding Process

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

International Journal of Advanced Research in Computer and Communication Engineering
Vol. 4, Issue 6, June 2015

In this algorithm, a secret image will be hidden in a cover image using LSB. Embedding operation is variable. The number of LSB bits used to embed could be vary from 1 bit to 8 bits. These numbers is used to add more security to the system because the receiver cannot extract the secrete image without knowing the number of bits of each channel used for embedding. This embedding sequence is chosen by the sender. An example of the embedding sequence is in the Fig.4.

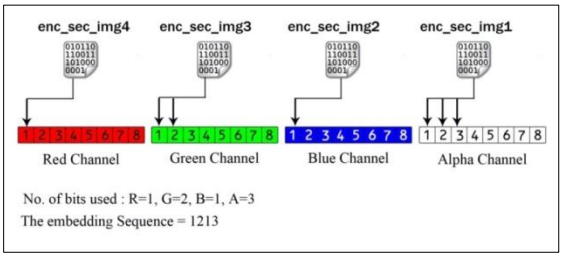


Fig.4 Example of embedding sequence

The steps for this algorithms are:

1. The secrete image is decomposed using Bit-plane slicing to 4 bit-planes if it is a gray scale image, and 12 bit-planes if it is a color image. Then convert the selected bit-planes into 1D arrays.
2. Encrypt the 4 1D bit-streams using a private key.
3. Extract Red, Green and Blue planes form cover image, and define the Alpha channel.
4. embedenc_sec_img1 into Alpha channel.

suppose the number of LSB bits can be used for embedding is N :

If $(1 \le N \le 8)$

(Embed N bits of enc_sec_img1 into each and every pixel of Alpha channel till message is not finished)

end

5. embedenc_sec_img2 into Blue channel.

If $(1 \le N \le 8)$

(Embed N bits of enc_sec_img2 into each and every pixel of Blue plane till message is not finished)

End

6. embedenc_sec_img3 into Green channel.

If $(1 \le N \le 8)$

(Embed N bits of enc_sec_img3 into each and every pixel of Green plane till message is not finished)

End

7. embedenc_sec_img4 into Red channel.

If $(1 \le N \le 8)$

(Embed N bits of enc_sec_img4 into each and every pixel of Red plane till message is not finished)

End

The Design flow of the extraction process is shown in Fig. 5

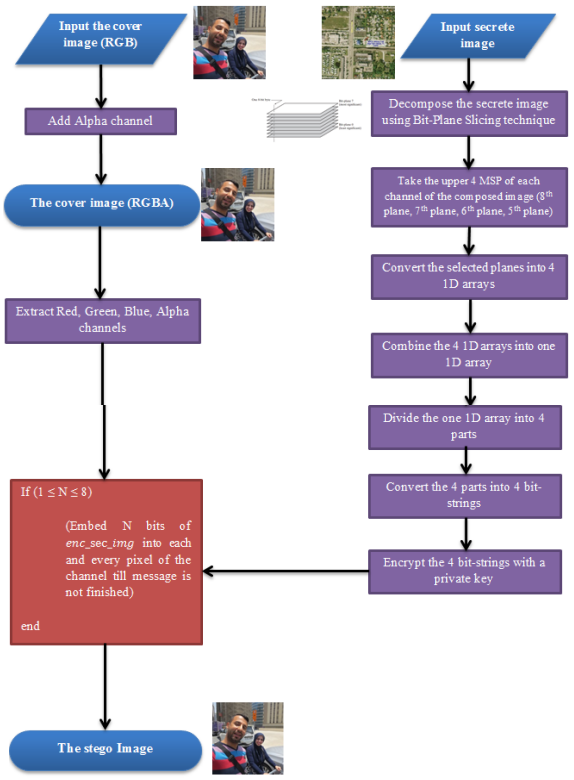


Fig.5 Design flow of hiding process

E. The Extraction Process

The extracting algorithm is the inverse of the embedding algorithms , as shown below:

1. Extract Alpha channel and Red, Green and Blue plane from RGBA stego image.
2. Use the embedding sequence to extract the encrypted secret bit strings (enc_sec_img) from each plane of the image. The original cover image is produced in this stage.
3. Use the secrete key to decrypt the secrete strings by XOR secrete key with encrypted bits.

$$\begin{aligned} \text{sec}_{str1} &= \{enc\_sec\_img1 \oplus \text{sec}\_key\} \\ \text{sec}_{str2} &= \{enc\_sec\_img2 \oplus \text{sec}\_key\} \\ \text{sec}_{str3} &= \{enc\_sec\_img3 \oplus \text{sec}\_key\} \\ \text{sec}_{str4} &= \{enc\_sec\_img4 \oplus \text{sec}\_key\} \end{aligned} \quad \ldots(16)$$

4. Combine all these bit-planes into one image to find the recovered secrete image Sec_im using following formula.

$$Sec_{im} = (\text{sec}_{str1} * 16) + (\text{sec}_{str2} * 32) + (\text{sec}_{str3} * 64) + (\text{sec}_{str4} * 128) \quad \ldots(17)$$

DOI 10.17148/IJARCCE.2015.4627

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

International Journal of Advanced Research in Computer and Communication Engineering
Vol. 4, Issue 6, June 2015

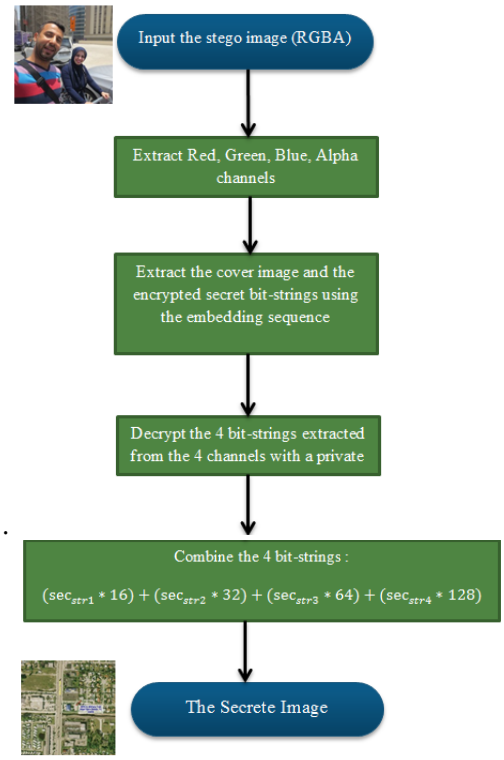


Fig.6 Design flow of the extraction process

III.SIMULATION AND RESULTS

A series of experiments have been conducted to show the effectiveness of the proposed technique. The efficiency of the proposed technique is measured by Five metrics which are:PSNR (Peak Signal-to-Noise Ratio), MSE (Mean Square Error), NCC (Normalized Cross Correlation), AD (Average Difference), and Histogram Analysis.

PSNR is usually measured in dB and given by :

$$PSNR = 10log_{10}\frac{(L-1)^2}{\frac{1}{M\times N}\sum_{i=1}^{M}\sum_{j=1}^{N}[St(i,j)-C(i,j)]^2}$$

Where: *N*: height of the two images (because the two images must be of the same size), *M*: width of the two images, *i and j* : row and column numbers, *L*: is the number of the gray scale levels in the two images, C(i,j): is the original image. St(i,j): is the stego image.

Typical PSNR values range between 20 and 40 dB [8].

Where MSE shows the mean square error between cover image C and stego image S.

$$MSE = \frac{1}{M*N}\sum_{i=0}^{m-1}\sum_{j=0}^{n-1}[S(i\ ,\ j)-C(i\ ,\ j)]^2$$

The Normalized-Cross Correlation is given by :

$$NCC = \frac{\sum_{i=0}^{m}\sum_{j=0}^{n}[S(i,j)*C(i,j)]}{\sum_{i=0}^{m}\sum_{j=0}^{n}[C(i,j)*C(i,j)]}$$

Average Difference of an image is given by :

$$AD = \frac{\sum M*N[1(m,n)-2(m,n)]}{M*N}$$

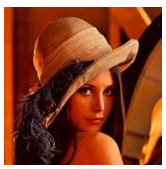
Large value of AD indicates that image has poor quality .

The histograms of the cover image and the stego image were found to show that the statistical properties of the cover image were not affected by changing some bits, so if the histogram of the stego is nearly equal to the histogram of the cover image, then this means that proposed system was good enough to avoid the attackers [9]. For testing the proposed method, Lena.jpg is chosen as an original cover image with size of 512*512 bit depth of 24 and colour system of RGB.Cameraman.bmp chosen as a secrete image with gray-level, 128*128 size, and 8 bit depth. The results are shown in the Table I :

Table I The results of embedding (128x128) gray-level secrete image into (512x512) RGB cover image

No. of bits	PSNR	MSE	NCC	AD	Bits per channel			
					R	G	B	A
4	50.544	0.2506	0.9996	0.0297	1	1	1	1
5	49.625	0.3171	0.9994	0.0464	1	1	1	2
6	48.312	0.2721	0.9995	0.03813	1	1	2	2
7	47.829	0.4728	0.9994	0.0515	1	2	2	2
8	47.457	0.4633	0.9994	0.0486	2	2	2	2
9	46.135	0.7828	0.9991	0.0753	2	2	2	3
10	44.037	0.6887	0.9992	0.0644	2	2	3	3
11	43.25	1.2283	0.9998	0.0036	2	3	3	3
12	42.692	1.2186	0.9998	0.00067	3	3	3	3

Fig.7 shows the original image, secrete image, stego-image, and the Histogram analysis of the cover and stego-image :



(a)

(b)

(c)

DOI 10.17148/IJARCCE.2015.4627

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

International Journal of Advanced Research in Computer and Communication Engineering
Vol. 4, Issue 6, June 2015

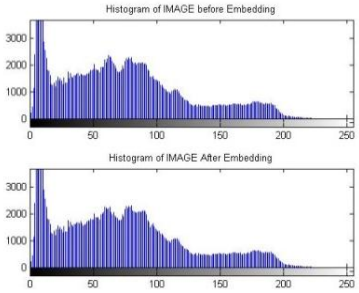
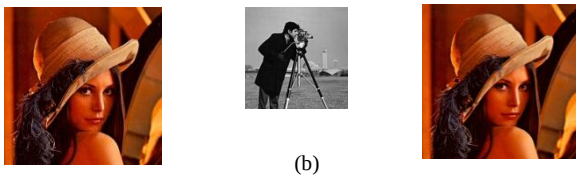


Fig.7(a) Original image, (b) secrete image(128*128),(c) stego-image, and the Histogram of Cover and Stego image

Table II shows the results of choosing cameraman.bmp gray-scale image with size of 256*256:

Table II The results of embedding (256x256) gray-level secrete image into (512x512) RGB cover image

No. of bits	PSNR	MSE	NCC	AD	Bits per channel			
					R	G	B	A
4	44.437	1.0925	0.9981	0.099	1	1	1	1
5	43.546	1.3656	0.9976	0.1674	1	1	1	2
6	42.332	1.1894	0.9979	0.1335	1	1	2	2
7	41.815	1.995	0.997	0.203	1	2	2	2
8	41.53	1.894	0.997	0.1923	2	2	2	2
9	40.239	3.0514	0.9964	0.2937	2	2	2	3
10	38.147	2.728	0.996	0.2477	2	2	3	3
11	37.1885	5.5061	0.998	0.0579	2	3	3	3
12	36.5	5.581	0.998	0.0488	3	3	3	3



(a) (b) (c)

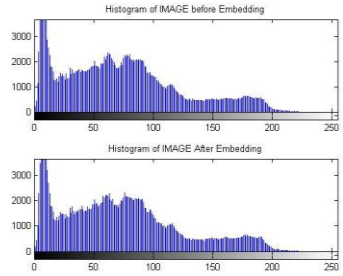


Fig.8 (a) Original image, (b) secrete image(256*256), (c) stego-image, and the Histogram of Cover and Stego image

Table III shows the results of embedding cameraman.bmp gray-scale image with the same size of cover image.

Table III The results of embedding (512x512) gray-level secrete image into (512x512) RGB cover image

No. of bits	PSNR	MSE	NCC	AD	Bits per channel			
					R	G	B	A
4	38.506	4.3362	0.992	0.2662	1	1	1	1
5	37.391	6.3766	0.9861	0.6857	1	1	1	2
6	36.449	5.9426	0.9866	0.599	1	1	2	2
7	35.897	9.464	0.9831	0.8937	1	2	2	2
8	35.647	9.409	0.9835	0.842	2	2	2	2
9	33.866	17.795	0.976	1.363	2	2	2	3
10	32.1654	16.981	0.9771	1.2217	2	2	3	3
11	31.319	26.3119	0.9818	0.7598	2	3	3	3
12	30.605	26.0438	0.9811	0.8022	3	3	3	3

Fig.9 shows the original image, secrete image, stego-image, and the Histogram analysis of the cover and stego-image :

(a) (b) (c) (d)

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

International Journal of Advanced Research in Computer and Communication Engineering
Vol. 4, Issue 6, June 2015

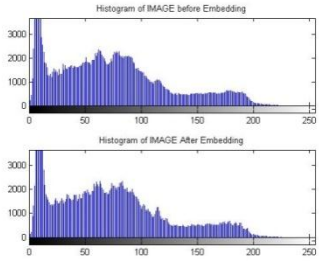


Fig.9 (a) Original image, (b) secrete image(512*512), (c) stego-image, and the Histogram of Cover and Stego image

Table IV shows the results of embedding indianman.bmp gray-scale image with size of 512*512 and data size bigger than the cover image to show the ability of this method to handle a high capacity embedding.

Table 1 The results of embedding a 257KB (512x512) gray-level secrete image into a 198KB (512x512) RGB cover image

No. of bits	PSNR	MSE	NCC	AD	Bits per channel			
					R	G	B	A
4	38.457	4.6872	0.9926	0.2615	1	1	1	1
5	37.2649	6.9768	0.9862	0.6817	1	1	1	2
6	36.2966	6.6748	0.9866	0.6201	1	1	2	2
7	35.7218	9.6385	0.9855	0.7038	1	2	2	2
8	35.1623	9.7728	0.9843	0.8263	2	2	2	2
9	33.3292	18.2306	0.9764	1.3499	2	2	2	3
10	31.7014	17.4939	0.9778	1.1836	2	2	3	3
11	30.884	27.664	0.9802	0.9206	2	3	3	3
12	30.4017	27.533	0.98169	0.7997	3	3	3	3

Fig.10 shows the original image, secrete image, stego-image, and the Histogram analysis of the cover and stego-image :

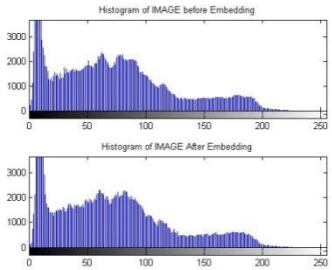


Fig.10 (a) Original image, (b) secrete image (512*512), (c) stego-image, and the Histogram of Cover and Stego image

IV. CONCLUSTION

In this work, a new data hiding technique presented, that allows hiding a color image (secret object) in another color image (cover object), where both images might be of same size or bigger, therefore achieving up to 100% embedding capacity. The stego image is very close to cover image in both objective and subjective tests. Statistical results show that the system has high invisibility.

Using Bit-Slicing technique compresses the secrete image, and this results in decreasing the total amount of data embedded. Also the attaching the alpha channel to the RGB image increases the bit depth of the image and this results in increasing the embedding range.As the results shows, Alpha channel can handle more bits than the other channels while maintain a good PSNR considering that the Alpha channel is the lowest byte of the RGBA pixel.

ACKNOWLEDGMENT

We wish to acknowledge H.O.D of Electronics and Communication engineering department of our college for their kind support for this project. We also thank our project guide and co-guide for highlighting our path and their gracious guidance. In last we like to thank all the friends who had given some valuable contribution for this system..

REFERENCES

[1] Shamim Ahmed Laskar, Kattamanchi Hemachandran," Secure Data Transmission UsingSteganography and Encryption Technique", International Journal on Cryptography and Information Security (IJCIS), Vol.2, No.3, September 2012.

[2] Veerdeep Kaur Mann, Harmanjot Singh Dhaliwal, " 32×32 Colour Image Steganography ", International Journal of Engineering Trends and Technology (IJETT), Volume 4, Issue 8, August 2013.

[3] Shuchi Sharma, Uma Kumari, " A High Capacity Data-Hiding Technique Using Steganography ", International Journal of Emerging Trends & Technology in Computer Science (IJETTCS), Volume 2, Issue 3, May – June 2013.

[4] Rafael C. Gonzalez,Richard E. Woods, " Digital Image Processing 3rd edition ", Prentice Hall, UpperSaddle River, 2008.

[5] Hedieh Sajedi ,Mansour Jamzad," Secure steganography based on embedding capacity", Springer-Verlag, International Journal of Information Security, volume 8, Issue 6, 2009.

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

International Journal of Advanced Research in Computer and Communication Engineering
Vol. 4, Issue 6, June 2015

[6] Rosanne English," Comparison of High Capacity Steganography Techniques", IEEE,International Conference of Soft Computing and Pattern Recognition, 978-1-4244-7896-2, 2010.

[7] S. K. Muttoo, Sushil Kumar," A Multilayered Secure, Robust and High Capacity Image SteganographicAlgorithm", IEEE,3rd International Conference onCommunication Systems Software and Middleware and Workshops, COMSWARE, 2008.

[8] Amer J. Sadiq,"Comparison Steganography in spatial domain of Image", Journal of Baghdad College of Economic Sciences, No.29,Baghdad, 2012.

[9] Zaynab Najeeb Abdulhameed, Prof. Maher K. Mahmood, " High Capacity Steganography based on Chaos and Contourlet Transform for Hiding Multimedia Data ", InternationalJournal of Electronics and Communication Engineering & Technology (IJECET), Volume 5, Issue 1, January 2014.