Securing Web Communication using Three Layer Image Shielding

Kamal Pradhan
Student, B. Tech in Computer Science and Engineering
Sambalpur University Institute Of Information Technology, Sambalpur, Odisha, India.
kamal.pradhan@suiit.ac.in

Gaurav Gohil
Student, Master of Computer Application
Sambalpur University Institute Of Information Technology,
Sambalpur, Odisha, India.
gourav.gohil@suiit.ac.in

ABSTRACT
Communication security has taken an important role with the advancement in digital communication. The difficulties in ensuring an individual's privacy has become increasingly challenging. Techniques such as digital watermarking, cryptography and Steganography are used for information hiding. This paper introduces a new Steganography algorithm to hide data inside images using three layer image shielding. Steganography is the art and science of hiding the existence of data in another transmission medium. It helps in achieving a secure and safe communication. The proposed algorithm uses spatial domain Steganography technique in the transformed color space. Here the three layers RGB (red, green, blue) of the cover image are transformed to HSV (hue, saturation, value) layers. The pixels of any two HSV layers are used to embed the message inside it. The remaining layer act as an indicator to store and retrieve the message from the other two layers efficiently. The final image is the stego image. Different sizes of data are stored inside the images and the PSNR (Peak signal-to-noise ratio) is also captured for each of the tested images. Based on the PSNR value of tested images, the stego image has a higher PSNR value.

Keywords
Steganography, Spatial domain, PSNR, Cover image, Stego image.

1. INTRODUCTION
The internet has revolutionized all forms of communication since the beginning of its existence and serves an important role in data transmission and sharing. Since the rapid growth of internet, information privacy and security have become the most important issues in today's world. Since the last 2 decades many information hiding techniques have been developed such as digital watermarking, Cryptography and Steganography. Watermarking is the process of embedding a message on a host signal. It has the additional requirement of robustness against possible attacks. A watermark can be either visible or invisible. Using digital watermarking, copyright information can be embedded into the multimedia data Information such the serial number, images or text with special significance can be embedded. The function of this information can be for copyright protection, secret communication, authenticity and distinguishing of data file, etc [1].
Cryptography is the art of hiding the contents of a message from an attacker, but it doesn't hide the existence of the message. Cryptography's main task is to ensure that, users are able to communicate securely over an insecure channel. This communication however must ensure the transmission's privacy and authenticity [2]. Steganography is the art and science of invisible communication. It is accomplished through hiding of information within other information, thus hiding the presence of the communicated information [3]. The word Steganography is derived from the Greek words "stegos" meaning "cover" and "grafia" meaning "writing" defining it as "covered writing". In image Steganography the information is hidden completely in images [4].The idea and practice of hiding information has a long history. In the pages of history the Greek historian Herodotus writes of a nobleman, Histaeus, who needed to communicate with his son in law in Greece. He shaved the head of one of his most trusted slave and tattooed the message onto the slave's scalp. When the slaves' hair grew back the slave was dispatched with the hidden message [5]. In the Second World War the micro dot technique was developed by the Germans [6]. Both Steganography and digital watermarking employ Steganography techniques to embed data covertly in noisy signals. But whereas Steganography aims for imperceptibility to human senses, digital watermarking tries to control the robustness as top priority. Steganography and cryptography are the art of hiding information without detection, both of them belong to the same family, cryptography scrambles a message so that it cannot be read. Steganography just hides it not to attract attention and this is the advantage that Steganography takes over cryptography.

This paper is structured in using the following format: Section 2 discusses the related works followed by our proposed algorithm in section 3. The experimental results of this algorithm are presented in section 4. Finally we conclude the work in section 6.

2. Related works
The image Steganography can be broadly divided into two categories namely spatial domain and frequency domain. In each of these categories we can have adaptive and dynamic methods. Adaptive methods are based upon image based statistics where as dynamic methods are message bit dependent [7]. Generally, for hiding information inside images least significant bit (LSB) method is used. This method does not increase the size of file but if size of information is increased the file fidelity degrades. Gutub et al [13] describes the pixel indicator technique where one channel is used to locate the channel to store data. There have been many statistical techniques developed to determine if an image has been subjected to LSB embedding [10] [11] [6]. Problems with existing methods that embed within the palette are that they do not take into account other important color models. Also, the information is limited and the hidden message can be destroyed by switching the order of the palette [9][8]. To overcome this we can use palette based images to embed

information inside them using different color models [8]. Palette-based images are used as cover images to provide a secure and fast transmission/storage over a communication system. Palette-based images are largely available on the Internet. Due to their abundance over the Internet, it is difficult to find a suspicious stego-image [8] [9]. A color image can be illustrated in a different color model [9].The purpose of color models is to organize colors in a standard form. Different models are used according to the user's need. These models are divided in two models: hardware oriented and color manipulation. The color models include RGB, CMY, CMYK, HSI, HSV, RGB and YIQ [8]. Palette based Steganography hides the Steganography message within the bits of the palette and the indices. Care must be taken while using this image file format ensuring that the number of colors is not exceeded. Examples of this form of embedding are BPCS and EzStego [9] but these algorithms are weak against visual attacks and steganlysis due to more distortion. Agaian and Perez [9] presented a new windowing technique for embedding messages in palette/color-map based images. This new method has the advantage of embedding secure data, within the index, the palette or both, using special sorting scheme. El-Emam [14], on the other hand, proposed a steganography algorithm to hide a large amount of data with high security. His Steganography algorithm is based on hiding a large amount of data (image, audio and text) file inside a color bitmap (bmp) image. According to his research, the image would be filtered and segmented where bit replacement is used on the appropriate pixels. Gandharba and Saroj[7], proposed an algorithm that divides the RGB image into 8 blocks and embeds the encrypted cipher text inside the 8 blocks in a regular pattern which provides an extra layer of security. Parvez and Gutub [15], proposed an algorithm that uses actual color of the channel to decide no of data bits to store. This approach leads to very high capacity with low visual distortion.

In this work, we propose a new Steganography algorithm which uses a palette based RGB image as our cover image. The RGB image is first transformed to HSV image and then Value and Saturation layers are divided into 4 non overlapping blocks. The information is hidden into two of the layers of HSV image in a spiral matrix form and the third layer act as the index or the pixel indicator of the stored data. The whole data is secured by stego key which is embedded in one of the block Hue or Saturation layer.

3. Proposed algorithm

In this section we propose an algorithm called three layer image shielding that hides large volumes of information inside an image with minimal degradation and high security. Here the three layers refer to a security protocol by which the stego image is generated. The first layer is our stego key layer which is common in all Steganography algorithms and this key gives us the access to the hidden information inside the images. The second layer is the encrypted code layer here the information is first converted into unique encrypted codes using a cryptography algorithm which can be decoded by the algorithm in receivers end. The third and the most important layer is our HSV palettes where the encrypted codes are embedded inside the two layers mainly saturation and value of HSV by changing a specific bit which uses spatial domain technique of Steganography. These layers prevent the image steganalysis and statistical attacks. The HSV image comprises of three layers Hue, Saturation and value. Hue describes the true color properties and Saturation describes strength or dominance of Hue. Value describes the overall intensity to how light or dark a color is [12]. Fig 1 shows the cylindrical model of HSV color space we can see here how the hue saturation and value change their properties. The value of Hue varies from 0o to 360o and the Saturation and value ranges from 0 to1.

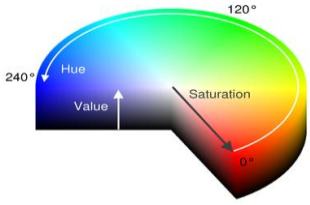


Fig 1: Cylindrical HSV color model

The information is stored in the Saturation and Value layer because minor changes here are schemes to be undetectable by the human visual system (HVS). The changes made in Hue layer affects the true color directly so we use the Hue layer for Pixel indication where minor changes are made to the bit. The saturation and value layer are divided into 4 non overlapping blocks of different sizes the blocks are S1,S2,S3,S4 and V1,V2,V3,V4 the information is stored in the 6 blocks S1,S2,S3 and V1,V2,V3 and the other two blocks S4 and V4 store the stego key, no of changed bits, the index of the pixels where our algorithm starts and stops and the meta information about encoded message. The Hue layer which is subject to minor change stores the index and act as an indicator for the pixel to be selected.

Our algorithm is divided into two parts as followed a: sender's end where the information is embedded into image and b: receivers end where the information is retrieved are discussed in sub sections 3.1 and 3.2.

3.1. Information embedding algorithm

- First the RGB image is converted into HSV image using color space transformation.
- The three layers Hue, Saturation and value are extracted from the HSV image these layers are in form of 2D matrix.
- The saturation and value layers are divided into 4 non overlapping blocks and the hue layer remains as it is. Fig 2.1 shows the three layers and how the blocks are divided.

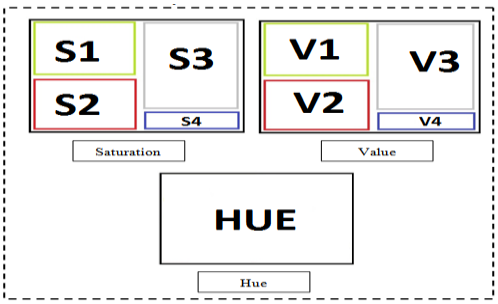
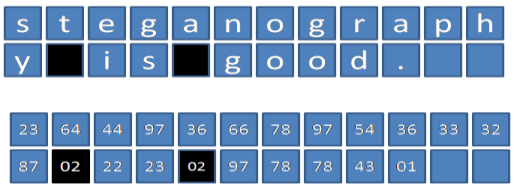


Fig 2.1: Different layers of HSV image after extraction

- The input message is now converted into encrypted codes using a cryptography algorithm. Let the input message be

  "Steganography is better".

Input message and encrypted codes.



The blue boxes show the input message and the black box refers to the space between two corresponding words the encrypted codes are decimal numbers.

- Once we have generated the encrypted code from the input message then we start embedding the encrypted codes into blocks of the Saturation and Value layer of the HSV image and add the corresponding index to the Hue layer.
- Let's take first 3 characters of the word "Steganography" i.e. "Ste" to be embedded in the Saturation layer and to add the index in Hue layer an encrypted code is only embedded when the value of the three layers is in between 0.1500 - 0.9500 else we leave them as they are because when the HSV image is converted to RGB after embedding the information to produce stego image the color transformation algorithm rounds off extreme value so to make our algorithm secure we embed the values in given range. Fig 3: shows how "ste" are embedded inside the pixels.

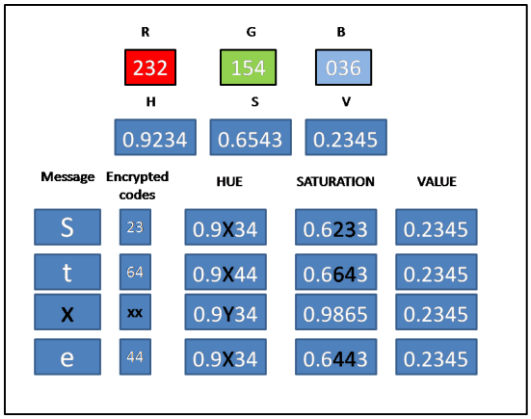
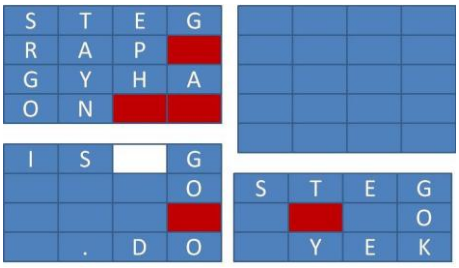


Fig 2.2: Embedding messages into HSV

In Fig 2.2 we can see pixel of RGB image after the transformation of image we get the corresponding pixels of HSV image. Now we start embedding the encrypted codes of corresponding message bit, first we check whether the Hue, Saturation and Value lies between the range 0.1500 - 0.9500 if the condition satisfies the code is embedded else the Value and Saturation are left blank and a false index 'Y' is added to Hue where 'Y' is a odd number. The code for 's' is 23 since the Hue, Saturation and Value lies in between 0.1500 – 0.9500 so the 2nd and 3rd significant bit of saturation layer is replaced with 23 and an index 'X' is added to the Hue layer where 'X' can is an odd no similarly 't' and 'e' is embedded in third case the Saturation range exceeds so we don't use it and add a false index 'Y' to the hue layer similarly the whole message is added to the Saturation and Value layer. The range of hue and saturation for every image is different and the ranges are stored in the V4 block of image.

- The messages are embedded into the six blocks S1, S2, S3, V1, V3 and V3 in a spherical order. The stego key is stored in the S4 block and the starting and closing index of the six blocks are stored in the V4 bock. Fig 4 shows how finally data is embedded.

Fig 2.3: final image after embedding



- After embedding completely the image is again converted to RGB image which is the output, stego image.

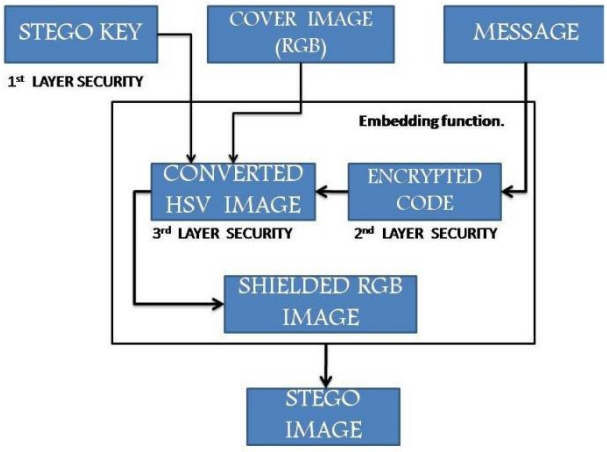


Fig 2.4: Flowchart of embedding algorithm at sender's end

3.2. Information retrieving algorithm

- First the stego image which is in RGB format is converted into HSV image using color space transformation.
- The three layers Hue, Saturation and value are extracted from the HSV image. The saturation and value layers are divided into 4 non overlapping blocks and the hue layer remains as it is. Fig 2.1 shows the three layers and how the blocks are divided.
- Then stego key is fetched from S4 block using the index stored in V4 block.
- The fetched stego key is then compared with user input key .If both key matches then using stego key the information is fetched from S1,S2,S3,V1,V2 and V3 blocks.
- Then the algorithm checks the starting and closing index of the S1, S2, S3, V1, V2 and V3 blocks which are obtained from V4 block.

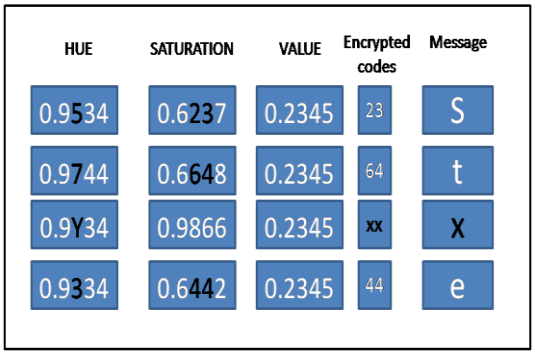


Fig 3.1: Retrieving of messages

Fig 3.1 shows how the algorithm first checks the range of Hue and Saturation and once the condition is satisfied we check the 2nd significant bit of Hue layer if the bit is odd we extract the 2nd and 3rd significant bit of saturation layer which is our encrypted code, similarly the whole encrypted code is extracted from different blocks in a spiral manner like shown in Fig 2.3.

- The encrypted codes are converted into message using our cryptography algorithm.

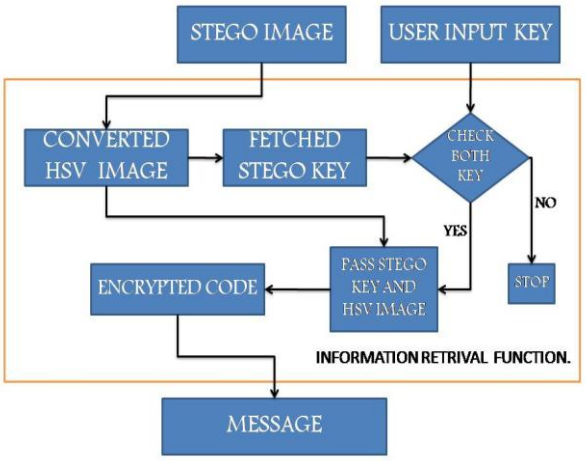


Fig 3.2: Flowchart of retrieving algorithm at receivers end.

4. The Experimental Results

The efficiency of all Steganography algorithms have to comply with same basic requirements. The requirements are Invisibility, Payload capacity, Robustness against statistical attacks and independent of file format. In this algorithm we have used two image formats BMP and PNG. The Peak Signal Noise Ratio (PSNR), Payload capacity of different image format is calculated and compared in two different formats finally the histograms of cover image and stego image are compared. We have carried out the experiment and implemented the above algorithm using MATLAB R2012b with two different images (a) Flower image (b) Pony image.

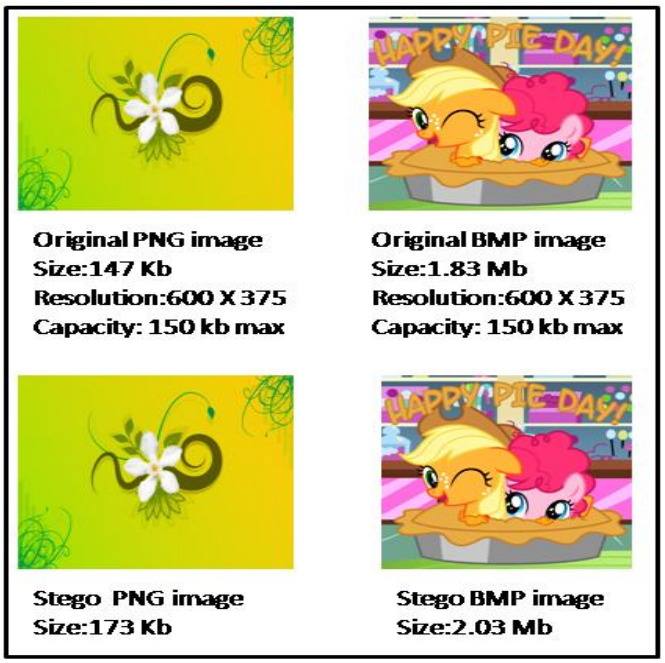


Fig 4: Comparison of cover and stego images

The above figure shows the comparison between the original PNG and BMP image with the stego BMP and PNG image it shows that the distortion by naked eyes between cover image and stego image is almost zero. The surfaces of both image shows no difference when viewed with naked eyes even though the size of stego image is slightly higher than the cover image.

We then test the algorithm using the PSNR (Peak signal-to-noise ratio). PSNR is a standard measurement used in Steganography technique in order to test the quality of the stego images. The higher the value of PSNR, the higher quality the stego image will have. If the cover image is C of size $M \times M$ and the stego image is S of size $N \times N$, then each cover image C and stego image S will have pixel value (x, y) from 0 to $M-1$ and 0 to $N-1$ respectively. The PSNR is then calculated as follows:

$$PSNR = 10.\log_{10}\left(\frac{MAX^2}{MSE}\right) \qquad (1)$$

where

$$MSE = \frac{1}{MN}\sum_{x=0}^{M-1}\sum_{y=0}^{N-1}(C(x,y) - S(x,y))^2$$

In equation (1) MAX represents the maximum possible pixel value of the image. For example, if the pixels are represented using 8 bits per sample, then the MAX value is 255.

If the stego image has higher PSNR value then stego image is more secure. Table 1.1 and 1.2 shows the comparisons between the both stego and cover image before and after embedding. The size of image increase after embedding of message since the increased size is negligible so both cover and stego images are alike, with the images in Fig. 4 we get PSNR value 86.1204 and 833217 when 1KB of data is stored in the pie image in both png and bmp format similarly we get 71.9543 and 73.4313 when 1kb of data is stored in flower image. The PSNR value decreases when the size of stored data increases. We get best results when we store 25KB of data which is equivalent to 6 pages and 6500 words in both images in all formats and get a PSNR value of 60.4147 and 61.3201 in pie image in both the formats respectively. Similarly we get PSNR values of 59.1625 and 60.1202 for the flower image.

Table 1.1- PSNR value for pie image

Size in bytes	Embedded Data Size in bytes	Stego Image Size in bytes	PSNR VALUE in Decibels
Pie image in Png format			
303104	25600	324608	60.4147
303104	1024	312320	86.1204
Pie image in Bmp format			
1918894	25600	2102231	61.3201
1918894	1024	1929841	83.3217

Table 1.2- PSNR value for flower image

Size in bytes	Embedded Data Size in bytes	Stego Image Size in bytes	PSNR VALUE in Decibels
Flower image in Png format			
150528	25600	324608	59.1265
150528	1024	312320	71.9543
Flower image in Bmp format			
674816	25600	2102231	50.1202
674816	1024	1929841	73.4313

Fig 6.1 and 6.2 show the color histogram plots the cover and stego images for all three channels. One more important thing to note from the histograms is that, our algorithm preserves the general shapes of the histograms. This feature of our algorithm makes it difficult to detect whether any data is hidden or not in the transmitted image.

Fig 6.1: Histogram for the cover image (pie.bmp).

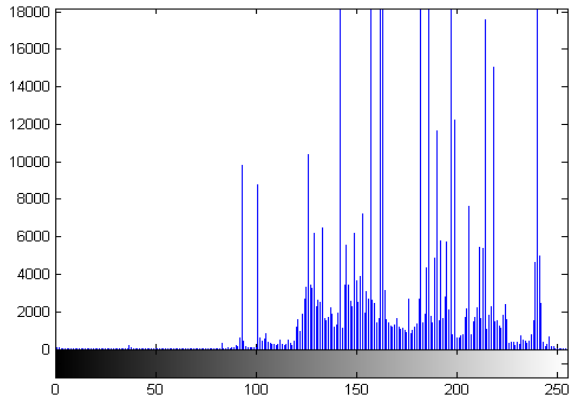
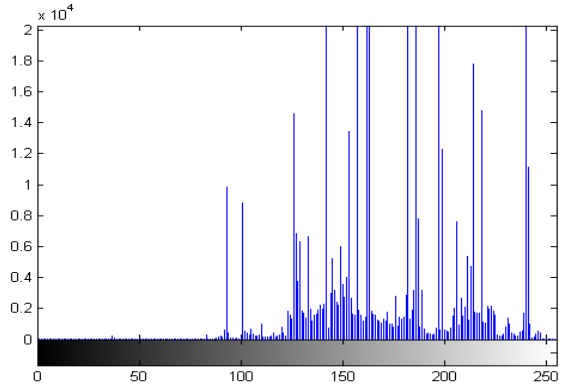


Fig 6.2: Histogram for the stego image (pie.bmp).



5. CONCLUSIONS

This paper proposes a new algorithm that provides three layered security using RGB image that makes it secure against Steganalysis and statistical attacks. It uses cryptography, Steganography and HSV palettes i.e. the three security layers. We have experimented and tested few images in various formats with the proposed algorithm; we found that the stego image does not have a noticeable distortion on it (as seen by the naked eyes). We embed the data in the HSV layers which increases the payload capacity of the image. We also get a high PSNR value so the algorithm is efficient to hide data inside images.

6. REFERENCES

[1]Stefan katzenbeisser, Fabien a. p. petitcolas, “Information hiding techniques for steganography and digital watermarking”, 2000, pp.

[2] Coron, J.-S., “what is cryptography? IEEE Security and Privacy, 2006. 4(1): p. 70-73”

[3] Akhil khare, Meenu kumarl, J Palla vi khare , “Efficient Algorithm for Digital Image Steganography”. Journal of information, knowledge and research in computer science applications.

[4] T. Morkel, j.h.p. eloff and M.s. olivier, ”An Overview of Image Steganography”,information and computer security architecture (icsa) research group.

[5]Jarno mielikainen, "lsb matching revisited", Signal Processing Letters, IEEE, Publication date: may 2006 Volume: 13, issue: 5, pp. 285- 287.

[6] Kanzariya nitin k. and Nimavat ashish v, “Comparison of Various Images Steganography Techniques”, vol 2 issue 1 jan13.

[7]Gandharba swain and Saroj Kumar lenka , “A novel Approach to RGB Channel Based Image Steganography Technique”. International Arab journal of e-technology, vol.2 no. 4 , june 2012.

[8] S. agaian1 and Juan p. perez2, “New Pixel Sorting Method for Palette Based Steganography and Color Model Selection.

[9] Mei-ching Chen, S. agaian, and C. L. philip chen,"Generalized collage steganography on images", IEEE, 2008.

[10]Chandramouli, r. & Memon, n. (2001). proceedings of ICPC ’01: IEEE International conference on image

processing. thessaloniki: Institute of electrical and electronics engineers computer society.

[11]Fridrich, j., Goljan, m., &Du, r. (2001). "Detecting lsb steganography in color and gray-scale images." IEEE multimedia, 8(4), 22-28.

[12] Wen chen1, Yun q. shi1, Guorong xuan2 , "Identifying computer graphics using hsv color model and statistical moments of characteristic functions."

[13] Adnan Gutub, Mahmoud Ankeer, Muhammad Abu-Ghalioun, Abdulrahman Shaheen, and Aleem Alvi, Pixel indicator high capacity technique for RGB image based Steganography, WoSPA 2008 – 5th IEEE International Workshop on Signal Processing and its Applications, University of Sharjah, Sharjah, U.A.E. 18 – 20 March 2008.

[14] N.N. El-Emam, Hiding a large amount of data with high security using steganography algorithm, Journal of Computer Science 3 (2007) 223-232.

[15]Mohammed tanver parver and Adnan Abdul Aziz-gutub, RGB intensity based variable-bits image Steganography, IEEE Asia-Pacific services computing conference.