

PIXEL PATTERN BASED STEGANOGRAPHY ON IMAGES

R. Rejani¹, D. Murugan² and Deepu V. Krishnan³

^{1,2}Department of Computer Science and Engineering, Manonmaniam Sundaranar University, India

E-mail: ¹rejani@gmail.com, ²dhanushkodim@yahoo.com

³Infosys Limited, India

E-mail: deepu_krishnan@infosys.com

Abstract

One of the drawback of most of the existing steganography methods is that it alters the bits used for storing color information. Some of the examples include LSB or MSB based steganography. There are also various existing methods like Dynamic RGB Intensity Based Steganography Scheme, Secure RGB Image Steganography from Pixel Indicator to Triple Algorithm etc that can be used to find out the steganography method used and break it. Another drawback of the existing methods is that it adds noise to the image which makes the image look dull or grainy making it suspicious for a person about existence of a hidden message within the image. To overcome these shortcomings we have come up with a pixel pattern based steganography which involved hiding the message within in image by using the existing RGB values whenever possible at pixel level or with minimum changes. Along with the image a key will also be used to decrypt the message stored at pixel levels. For further protection, both the message stored as well as the key file will be in encrypted format which can have same or different keys or decryption. Hence we call it as a RGB pixel pattern based steganography.

Keywords:

Data Protection, Steganography, Stegoimage, Cover Image, LSB, MSB, RGB

1. INTRODUCTION

The communication technologies around us has grown at a great pace in recent times. For exchanging of data/ information these days everyone is depending of high speed computer networks like internet which is quite unprotected and information can get exposed. Vast amount of personal data is often collected, used and transferred to third party organizations for a variety of reasons. Hence data security is becoming a serious problems in data communication via Internet or any other media. We can use Steganography or Cryptography to protect sensitive data. Steganography is often considered better than cryptography because the intended secret message does not attract attention to itself for scrutiny.

Steganography is the act of embedding information in a given media called covering media without making any visible changes in it. The goal is to hide an embedded file within the cover media such that the embedded file's existence is concealed. Image based steganography uses images as the covering media. Several methods have been proposed for image based steganography, LSB being the simplest one. Steganography plays the central role in secret message communication. Different message hiding techniques have been developed and implemented in the past using audio/video files, digital images, and other medias.

A digital image is a collection of data/information about the pixels in it. It is a large collection of data. This is large compared to the message we want to hide. Hence we always prefer a digital

image for covert communication. We can make use of its innocence for hiding data.

Each pixel is a combination of RGB i.e. is (Red, Green, and Blue). A 24-bit bitmap will have 8 bits representing each of the three color values (red, green, and blue) at each pixel. It makes a wide variety of colors. Since the data is big, any small change in the pixel intensity doesn't make any noticeable change. Also human visual system cannot identify the small changes in the pixel. Maintaining picture quality is an important for protection of the message. Enhancing this we use different techniques.

2. RELATED WORK

In [3] one type of LSB based RGB intensity steganography technique proposes an improved LSB (least significant bit) based Steganography technique for images. It deals with an embedding algorithm for hiding encrypted messages in nonadjacent and random pixel locations in edges and smooth areas of images. At first it encrypts the secret message, then detects edges in the cover-image by using improved edge detection filter. After that the message bits are embedded in the least significant byte of randomly selected edge area pixels and 1st, 3rd and 4th LSBs of red, green, blue combinations respectively across randomly selected pixels across smooth area of image. Some other types of LSB steganography techniques based on least bits are explained on [17]. And some similar technique is implemented in [16] MSB based steganography, it is light variation from LSB steganography.

In RGB Intensity Based Variable-Bits Image Steganography [4] describes new algorithm for RGB image based steganography. This algorithm introduces the concept of storing variable number of bits in each channel (R, G or B) of pixel based on the actual color values of that pixel: lower color component stores higher number of bits. Secure RGB Image Steganography from Pixel Indicator to Triple Algorithm-An Incremental Growth [1] introduces two methods of RGB image steganography one is pixel indicator technique and other is triple-A algorithm. They uses the same principle of LSB, As the secret is hidden in the least significant bits of the pixels, better randomization in selection of the number of bits and color used. This randomization increase the security of the system and it also increase the capacity. Some other types of RGB based steganography techniques are implemented in [1, 2, 11, and 14]. These techniques can be applied to RGB images where each pixel is represented by three bytes to indicate the additive values of red, green, and blue.

Paper [5] designs a steganography algorithm which not only hide the message behind the image but also provide more security than others. For the purpose of security, encryption technique is used with a user-defined key. In that paper, message is hide into

an image in the form of an image that is using image generation method message is converted into the image of predefined format and then by using designed algorithm that image will hide into the cover image. RGB image format is used to improve the quality of the stego image. At last that RGB image will saved as BMP image file so that no lossy compression can occur and the original message do not destroy and can be extract as it is.

In Pixel Indicator High Capacity Technique for RGB Image Based Steganography [6] give the ideas from the random pixel manipulation methods and the stegokey ones techniques are merged, which uses the least two significant bits of one of the channels to indicate existence of data in the other two channels.

3. EXISTING TECHNOLOGIES

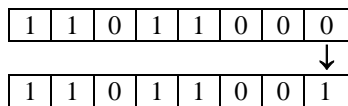
Hiding of data inside an image is simply called steganography. A lot of steganography techniques are used frequently to cover an information, [7] is an image steganography technique and [8] is an JPEG based steganography technique. Several spatial domain techniques are considered. In that the easiest method is LSB (Least Significant Bit) Steganography. In this paper for discussion we have considered LSB steganography and RGB steganography. There exists two types of LSB steganography methods – LSB1 Steganography and LSB2 steganography. RGB Steganography also have a lot of variations similarly.

3.1 LSB-1 STEGANOGRAPHY

This is the simplest of the steganography methods based in the use of LSB, and therefore the most vulnerable. Embedding process consists of the sequential substitution of each Least Significant Bit (LSB-1) of the image pixel for the bit message [17][2] [9] [12]. For its simplicity, this method can camouflage a great volume of information. The guidelines are given below:

- Step1:** Convert the data from decimal to binary.
- Step 2:** Read cover image.
- Step 3:** Convert the cover Image from decimal to binary.
- Step 4:** Break the byte to be hidden into bits.
- Step 5:** Take first 8 byte of original data from the cover Image.
- Step 6:** Replace the least significant bit by one bit of the data to be hidden.

First byte of original information from the Cover image:
E.g.:-1 1 0 1 1 0 0 0
First bit of the data to be hidden: 1
Replace the least significant bit



This process will be continued for first 8 byte of data and conceal the first byte of data.

- Step 7:** Continue the step 6 for all pixels.
- Images after embedding data using LSB-1 Steganography.



Fig.1. Cover Image

Fig.2. Stego Image

3.2 LSB –2 STEGANOGRAPHY

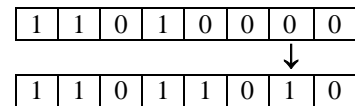
In LSB-2 Steganography the data embedding process is slightly different. It alters the 2nd bit from right for all pixels [17]. The algorithm is as follows:

- Step1:** Convert the data from decimal to binary.
- Step 2:** Read Cover image.
- Step 3:** Convert the Cover Image from decimal to binary.
- Step 4:** Break the byte to be hidden into bits.
- Step 5:** Take first 8 byte of original data from the Cover Image.
- Step 6:** Replace the least significant bit by one bit of the data to be hidden.

First byte of original information from the Cover Image:
E.g.:- 1 1 0 1 1 0 0 0

First bit of the data to be hidden: 1

Replace the least significant bit



This process will be continued for first 8 byte of data and conceal the first byte of data.

- Step 7:** Continue the step 6 for all pixels.

After applying the algorithm the images are



Fig.3. Cover image

Fig.4. Stego image

3.3 RGB STEGANOGRAPHY

To a computer an image is an array of numbers that represent light intensities at various points (pixels) these pixels makeup the image’s data. Digital images are normally stored in either 24-bit (RGB) or 8-bit (Grayscale) files. A 24-bit image provides the most space for hiding information; however it can be quite large (except JPEG images). All colors are derived from three primary

colors: red, green, and blue. Every primary color is represented by one byte i.e. each pixel represents a combination of (R, G, B).

Different RGB based algorithms are used [10] [11] etc for steganography. Each have advantages and disadvantages from the existing techniques. Dynamic RGB based approach [11] is used to change the least significant bits of pixel values (3) or sometimes (4) of the rearrangement of colors to create parity bit patterns or least significant bit which correspond to the message being hidden. Also variable bit steganography technique is used in RGB based steganography [4].

Image after embedding the data using RGB steganography technique.



Fig.5. Cover image

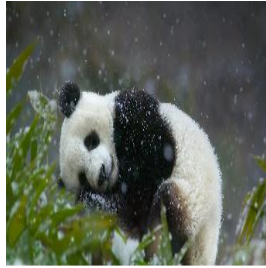


Fig.6. Stego image

4. PROPOSED TECHNIQUE

The proposed a technique in this paper is RGB pixel value based steganography method. The specialty of this algorithm is that we do not change the pixels like other steganography algorithms except if it is absolutely needed. To a computer an image is a collection of data/information that represents light intensities at various points (pixels) these pixels making up the image's raster data. All Digital images are typically stored in either (24-bit) RGB or (8-bit) known as Grayscale files. A 24-bit image provides the most space for hiding information; however it can be quite large (with the exception of JPEG images). All color combinations are derived from three primary colors - red, green, and blue. Each of this primary color is represented by one byte.

Because RGB values are all represented by numbers, we can make use of this numbers to represent text using a modbit algorithm. So what is a modbit algorithm? A modbit algorithm is very similar to the Luhn mod n algorithm. Traditionally a Luhn mod algorithm is widely used for generating checksum formula for validating credit card numbers, IMEI numbers etc. As part of this paper we have taken the concept used in Luhn mod N algorithm [18] but for finding the pixels which can represent a character. Each character from the input text will be mapped to a set of numbers and this mapping will be maintained internally in the Stegano program [15]. For example letter 'a' could be represented by digit 10, letter 'b' could be represented by digit 12 and so forth. During the encryption process the Stegano program will scan the image and will add the RGB values, divide it and find the mod value. If mod matches the character, that location in the image could be used to represent the character.

But then problem arises on how will we be able to store the location of a pixel which can identify a character? We can either store it in a separate text file or make it as part of the image

metadata itself. To make it easy in this paper we are proposing to make it part of the image metadata itself.

Finally there will be also cases when we may not find a pixel in an image that might not be able to represent a particular character. In these cases the work around is to alter some of the places in an image to a nearest possible pixel such that it can represent the character and at the same time will be not identifiable by human eye.

4.1 ENCRYPTION PROCESS

Step 1: The application prompts for the text and image from the sender who wants to hide the message.

Step 2: Steganographic program encrypts the text using DES or RSA or any other encryption algorithm.

Step 3: Steganographic program analyses the image to find the pixel value of all the pixels within the image.

Step 4: Steganographic program uses the unique RGB modbit method to find out whether each letter of the message can be represented in the image and records the position to a field in the image metadata itself. For calculation of modbit the program adds the RGB values of each pixel and divides it to get the mod. If the mod value matches with that represented for the character internally, the position for that character is recorded.

Step 5: If the image does not have pixel values to represent a particular character, the steganographic program finds and changes a pixel that almost matches with the image pixel and which can represent the character of text.

Step 6: Finally when all the pixels which can be identified on the image and its position is recorded along with the image metadata, the user is informed that the encryption part is complete.

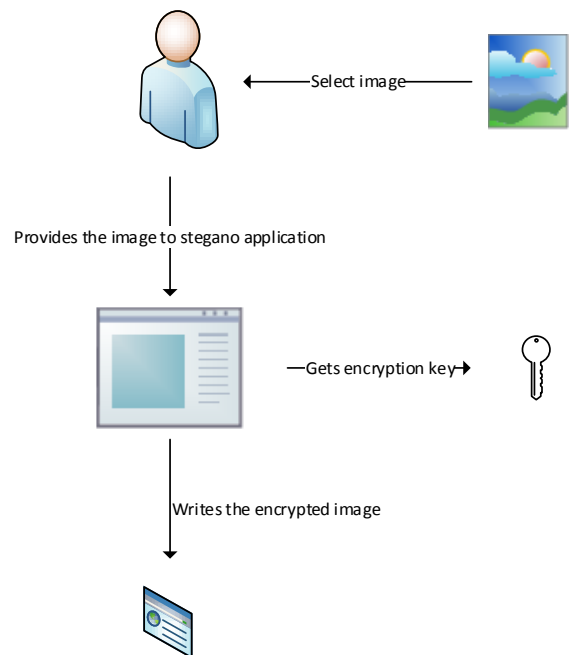


Fig.7. Data Embedding process

```

For all characters in the message text
{
  for all pixels in the image (all rows and columns)
  {
    Add the RGB values and find mod by division.
    Compare with modbitvalue with the internal table of
    values for the character maintained inside the stegano
    program.
    if values match then encrypt the pixel position and
    add it to the image meta data and exit this loop continue
    with next character
    else
      Continue with next pixel
    end if
    If none of pixel can represent the text change a pixel
    towards the edges to the nearest value which can
    represent the character and store it.
  }
}
    
```

4.1.1 Embedding Algorithm:

The advantage of this proposed technique is that it will not degrade the image quality as it depends on the pixel values. Hence the covering image and the stego image will not have any visual difference and will also be prone from any sort of attacks.

4.2 DECRYPTION PROCESS

- Step 1:** The receiver opens the image.
- Step 2:** The steganographic software asks for key to decrypt the image file.
- Step 3:** Steganographic software decrypts the metadata first and finds the pixel positions.
- Step 4:** Using the pixel positions, get the RGB values and decodes by reverse modbit and finds the corresponding encrypted text.
- Step 5:** Decrypt this text and provide back the message to the user.

```

Get the key
For all characters in the image metadata
{
  For all pixels in the image
  {
    Find the modbits from the pixels positions
    specified
    Decrypt the modbits
    Display the character continue with next
    character
  }
}
    
```

4.2.1 Extracting Algorithm:

Since the reverse modbit algorithm is hidden within the software only this decryption program will be able to decrypt the

message. Any other method of trying to get the text will result in failure.

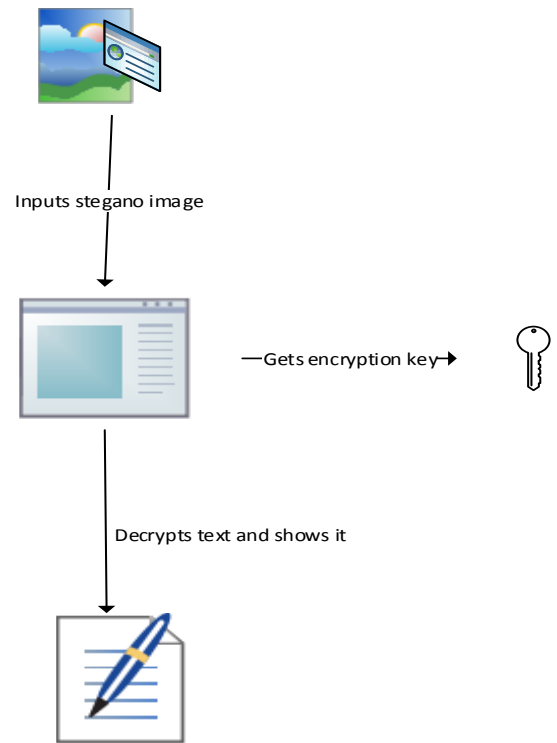


Fig.8. Data extraction process

5. EXPERIMENTAL ANALYSIS

To demonstrate the algorithm we developed a small application in .net. The first step is to encode the text to be hidden into the image which needs to be used. The application allows for both encoding and decoding the text to be hidden and show the o/p to the user. The application screens below show each of the steps of encryption and decryption processes.



Fig.9. Input the information to be hide and select the cover image

The information to be hide and the cover image can be selected through this input screen. The data is encoded and embed into the image. The pixel position information is encrypted and stored along with metadata of the image which holds the information about pixels.

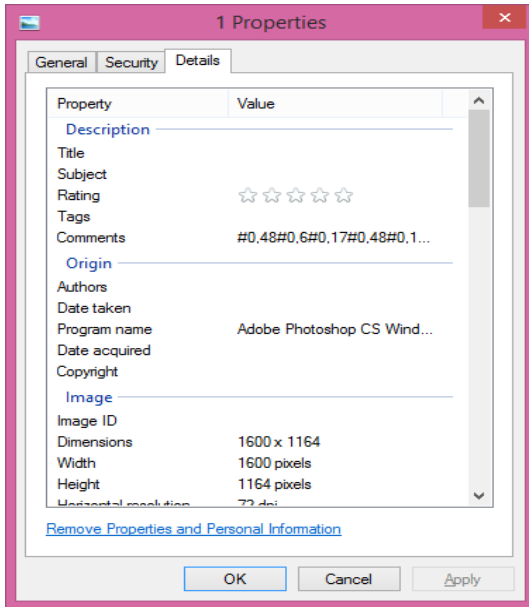


Fig.10. Information is stored in metadata of image

In this example we have used the comment field to store the encrypted pixel location information. But it can be any field in the metadata or the algorithm can also be adapted to use multiple fields in the metadata.

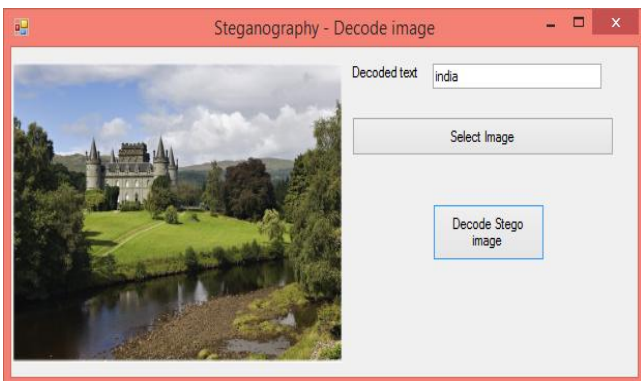


Fig.11. Decoded message is displaying

For decoding the text, we can invoke the decode screen which can select a stegano image and allow you to decode the message and display result on the screen.

The input image is given below:



Fig.12. Image Given

The resultant image after embedding the secret information is given below.



Fig.13. Resultant image

6. PERFORMANCE ANALYSIS

To check whether the method proposed in this paper is superior to the other methods, we decided to compare it against some of the other existing steganography methods.

PSNR value of the encoded image (Peak Signal to Noise Ratio) is normally calculated to find the noise of the image.

For calculating the PSNR value we have to find the MSE (Mean Squared Error) first using the formula,

$$MSE = \frac{1}{mn} \sum_{i=0}^{m-1} \sum_{j=0}^{n-1} [I(i, j) - K(i, j)]^2 \tag{1}$$

The PSNR is defined as:

$$\begin{aligned} PSNR &= 10 \cdot \log_{10} \left(\frac{MAX_I^2}{MSE} \right) \\ &= 20 \cdot \log_{10} \left(\frac{MAX_I}{\sqrt{MSE}} \right) \\ &= 20 \cdot \log_{10}(MAX_I) - 10 \cdot \log_{10}(MSE) \end{aligned} \tag{2}$$

where, MAX_I is Maximum possible pixel value of the image.

The minimum value of MSE denotes that both the given images are having almost same quality. Higher the PSNR value means the reconstruction is of higher quality. For all type of color images with RGB values per pixel the definition of PSNR is the same except the MSE is the sum over all squared value differences divided by image size and by three. At the same time, for color images the image is converted to a different color space and PSNR is reported against each channel of that color space.

The insertion capacity of image can be tested using the PSNR ratio. For the experiment, the standard color images of 512×512 (786,432byte) have been used and inserted information was tested with text and in the above mentioned techniques.

In each image same amount of data is inserted and checked. It is shown in the below Table.1.

Since the MSE value is zero the PSNR value goes to infinity in this case.

As we can see the PSNR value does not show any difference in the new technique used. Hence we can effectively store data inside any image without any noise. The image quality can be





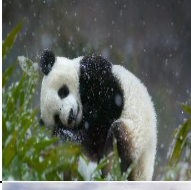
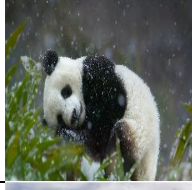
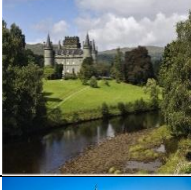





retained as the same. If anybody checks the stego image with an original image also nobody will not get any indication of any type of attacks. The only drawback of this method is that for encryption the time taken is more than the other methods. However this can be reduced to an extend by enhancing the algorithm in future. This will ensure maximum protection.

7. CONCLUSION

The paper proposed a new technique for image based steganography. It presents an improved steganography method for embedding secret message bit in image metadata fields based on the RGB values and the position of the pixels. The image

pixels will be changed only for characters where the algorithm cannot find a pixel which can represent it. Since only the metadata is modified, the stego image looks exactly the same as original image or at max it will be very difficult to identify the changes for the human eye. Only the size of the stego image will increase slightly however in our test cases this has been found to be comparable with other steganography methods. This research was aimed towards the development of a new and improved data hiding technique based on RGB based steganography without changing the image. Some of the possible application areas include transmitting small secret messages, using an image as a password token by encrypting and hiding password using this technique, simply adding a hidden signature to an image etc.

Table.1. Comparison among the steganography techniques along with the time taken for encryption

Technology Used	Cover Image	Stego Image	Size of cover Image	Size of stego Image	Data Inserted	Size of Data	PSNR Ratio	Time taken for encoding
LSB-1 Steganography			37.5 KB	147 KB	India	40 bits	35.46 db	< 1 sec
LSB-2 Steganography			29.1 KB	31.0 KB	India	40 bits	41.30 db	< 1 sec
RGB steganography			193 KB	257 KB	India	40 bits	28.91db	< 1 sec
The Proposed Algorithm			432 KB	434 KB	India	40 bits	1.#INF	3 sec
The Proposed Algorithm			881 KB	882KB	India	40 bits	1.#INF	4 sec
The Proposed Algorithm with modification to pixels			57KB	60KB	India	40 bits	45.49db	3 sec

Advantages of the new algorithm:

- 1) The image is virtually not changed. Hence there is no visible difference at all
- 2) The image size will increase slightly to store the additional metadata
- 3) It can be easily shared using any method
- 4) Noise is less compared to any other technique
- 5) Comparatively large amount of data can be stored without changing picture quality
- 6) Nobody can guess about any type of attacks inside the image
- 7) More secure than any other RGB methods.

REFERENCES

- [1] Namita Tiwari and Madhu Shandilya, "Secure RGB Image Steganography from Pixel Indicator to Triple Algorithm-An Incremental Growth", *International Journal of Security and Its Applications*, Vol. 4, No. 4, pp. 53-62, 2010.
- [2] Koyi Lakshmi Prasad and T. Ch. Malleswara Rao, "A Novel Secured RGB LSB Steganography with Enhanced Stego-Image Quality", *International Journal of Engineering Research and Applications*, Vol. 3, No. 6, pp. 1299-1303, 2013.
- [3] Mamta Juneja and Parvinder S. Sandhu, "An Improved LSB based Steganography Technique for RGB Color Images", *2nd International Conference on Latest Computational Technologies*, pp. 10-14, 2013.
- [4] Mohammad Tanvir Parvez and Adnan Abdul-Aziz Gutub, "RGB Intensity Based Variable-Bits Image Steganography", *IEEE Asia-Pacific Services Computing Conference*, pp. 1322-1327, 2008.
- [5] Babita and Ayushi, "Secure Image Steganography Algorithm using RGB Image Format and Encryption Technique", *International Journal of Computer Science and Engineering Technology*, Vol. 4, No. 6, pp. 758-762, 2013.
- [6] Adnan Abdul-Aziz Gutub, "Pixel Indicator Technique for RGB Image Steganography", Available at http://faculty.kfupm.edu.sa/COE/gutub/Publications/J_m_per.pdf
- [7] Hassan Mathkourand, B. Al-Sadoon and Ameer Touir, "A New Image Steganography Technique", *IEEE 4th International Conference on Wireless Communications, Networking and Mobile Computing*, pp. 1-4, 2008.
- [8] Qingzhong Liu, Andrew H. Sung, Zhongxue Chen and Xudong Huang, "A JPEG-Based Statistically Invisible Steganography", *Proceedings of the Third International Conference on Internet Multimedia Computing and Service*, pp. 78-81, 2011.
- [9] R. Chandramouli and Nasir Memon, "Analysis of LSB Based Image Steganography Techniques", *Proceedings of the IEEE International Conference on Image Processing*, Vol. 3, pp. 1019 -1022, 2001.
- [10] Amir Farhad Nilizadeh and Ahmad Reza Naghsh Nilchi, "Steganography on RGB Images Based on a Matrix Pattern using Random Blocks", *I. J. Modern Education and Computer Science*, Vol. 4, pp. 8-18, 2013.
- [11] Ep Kaur, Surbhi Gupta, Parvinder S. Sandhu and Jagdeep Kaur, "A Dynamic RGB Intensity Based Steganography Scheme", *World Academy of Science, Engineering and Technology*, pp. 833-836, 2010.
- [12] Mamta Juneja and Parvinder Singh Sandhu, "Designing of Robust Image Steganography Technique Based on LSB Insertion and Encryption", *Proceedings of the IEEE International Conference on Advances in Recent Technologies in Communication and Computing*, pp. 302-305, 2009.
- [13] Ankit Chaudhary, Jaldeep Vasavada, J. L. Raheja, S. Kumar and M. Sharma, "A Hash Based Approach for Secure Keyless Image Steganography in Lossless RGB Images", *The 22nd International Conference on Computer Graphics and Vision*, pp. 80-83, 2012.
- [14] K. Suresh Babu, K. B. Raja, K. Kiran Kumar, T. H. Manjula Devi, K. R. Venugopal and L. M. Patnaik, "Authentication of Secret Information in Image Steganography", *IEEE Region 10 Conference TENCN*, pp. 1-6, 2008.
- [15] R. Rejani, D. Murugan and Deepu V. Krishnan, "Stegano DB- A Secure Database Using Steganography", *ICTACT Journal on Communication Technology*, Vol. 4, No. 3, pp. 785-789, 2013.
- [16] Chen Ming, Zhang Ru, NiuXinxin and Yang Yixian, "Analysis of Current Steganography Tools: Classifications and Features", *IEEE International Conference on Intelligent Information Hiding and Multimedia Signal Processing*, pp.384 - 387, 2006.
- [17] A. E. Mustafa, A. M. F. ElGamal, M. E. ElAlmi and B. D. Ahmed, "A Proposed Algorithm For Steganography In Digital Image Based on Least Significant Bit", *Research Journal Specific Education Faculty of Specific Education Mansoura University*, pp. 752-767, 2011.
- [18] Wikipedia-http://en.wikipedia.org/wiki/Luhn_mod_N_algorithm.