A Symmetric Steganography with Secret Sharing and PSNR Analysis for Image Steganography

Ajit Singh and Upasana Jauhari

Abstract : Data transmission across networks is a common practice as per the development of Internet and multimedia technologies that grows exponentially today. The paper presented here is concerned with the secret sharing of message by hiding it in an image using the most commonly used LSB (Least Significant Bit) technique and later the calculations has been done for the PSNR (Peak Signal to Noise Ratio) of image in MATLAB. Steganography aimed at hiding the data invisibly within any media (image, audio, and video) so that it should be unnoticeable to the unintended person. Here the message to be sent is first encrypt with a key which enhance the security of message to another level.

Keywords- Symmetric Encoding, Digital Steganography, LSB embedding, PSNR

1 INTRODUCTION

While exchanging data electronically, the privacy and secrecy of message is of primarily concern. The encryption, transforming message (plain text) into cipher text (encoded form) and decryption, a reverse process, plays an important role in concealing the confidentiality of the message. The message is first encrypted with a key during a encryption process and then hiding it in available format (here used image). Thus sending an encrypted message increase the security level of the message. Once received the message need to be decrypt using same key [1] which implies the concept of symmetric key steganography in which the key is symmetric for both sender and receiver. The paper introduce an approach which enhance the security of data by first encoding it , hiding in cover medium and sent it to the intended recipient. A general steganography scheme is shown in figure 1:

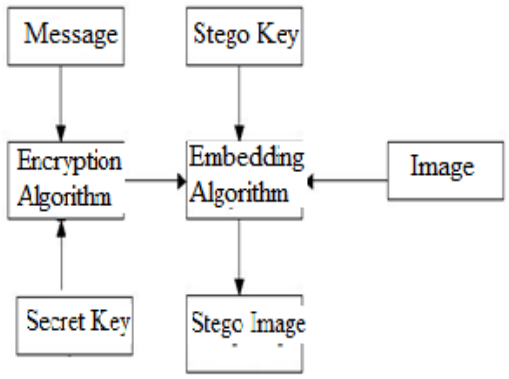

Figure 1: Steganography Process

We proposed a work in which the message needs to be sent is first encrypted using ASCII codes manipulation and a random string generated for the message which act as a key. As for decryption, the receiver needs three things- the message, the key, the random generated string which is an add-on to the security of message. Later the image containing message is analyzed for PSNR (Peak Signal to Noise Ratio) so that the changes in the image could not be compromised.

The paper is constructed in the sections where section 2 describe the symmetric encoding in which the data is encrypted using ASCII codes with symmetric key, section 3 defines the Digital Steganography in brief, section 4 describe the LSB(Least Significant Bit) embedding for

message hiding in image, section 5 describe the PSNR formulation, section 6 is implementation, section 7 is conclusion part.

2. SYMMETRIC ENCODING

Data encryption is just the mechanism for making the information unintelligible by making it unreadable without knowing the strategies being applied to it. Here the message is encrypted using a key which is symmetric which means that the encryption and decryption is carried out by the same key [1]. The general scenario is shown as:

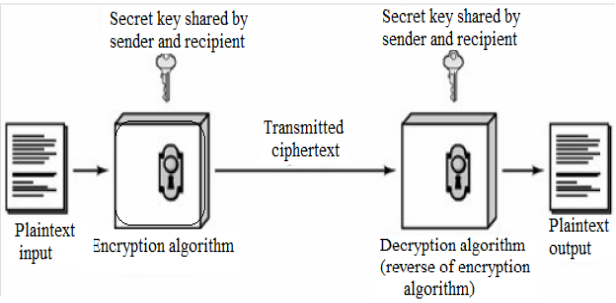

Figure 2: A symmetric encryption/decryption model

The basic operation is applied on the ASCII codes [2] which is generated and stored in file. The ASCII character encoding or a compatible extension is used on nearly all common computers, especially personal computers and workstations. ASCII codes [2] include definitions for 128 characters: 33 are non-printing, 94 are printable characters (the space is not printable). The representation of each and every character is with the seven bits (b7 to b1) e.g. the representation of A is (1000001) =65, similarly other characters are coded like this.
The encryption process applied on the input text is shown below:

2.1 Encryption Process:
During first phase of the algorithm

1. Generating the random strings and saving it into text file (in this case, creating 1000 random string)
2. Input the Text message for the encryption.
3. Generating the key for the text.
   a) Getting all the lines from the entered text and inserting them into the arrLines string array.
   b) Lines will be separated by the "." operator.
   c) From the arrLines array, arrWord array will be filled by all the words of the specific lines.
   d) By working on specific lines and on all the words of the lines, start the encryption process.
   e) Getting the character of the word one by one for the processing.
   f) Filling the arrSeq array that have the range for all the character (symbols, integers, alphabets)
   g) Convert the incoming character into its ascii value.
   h) Find the range corresponding to the incoming char. Ascii value by using arrSeq array.
   i) By using the range, generate a random integer between that range and fetch thats no's position random string from the random string txt file. Ex – for the asciii value 65, 651 – 660 is the range. Let 653 is the random no. generated, then find the random string placed on the 653th place in the random string text file.
4. After getting the random ascii string of that character, add four parameter in that.
   Four parameters are:
   i. linoNo (line number frm which that word relates),
   ii. posOfW (position of the word in that line),
   iii. lengthOfW (length of that word) and
   iv. posInWo (position of character in that word).

In 2nd phase, apply some mathematical operation in that encoded string.

1. Now key comes into play, store the entire individual integer in the arrKey array and all the individual ascii value into the arrCodes array.

2. In this case here using 5 length long integer value and performing respective mathematical operation on each value:

a. Ascii + first integer

b. Then multiplied by second integer

c. Then Ascii – third integer * 20

d. Then Ascii- fourth integer*5

e. Then Ascii-fifth integer

3. Convert into the symbol whose value is between 0 and 255 after applying operation and leave remain as it is.

4. The data retrieved after this is in corresponding encrypted form which needs to be store in separate file.

2.2 Decryption Process:

Following is the decryption process which involves the use of same key as generated during encryption process.

1. Using the key of length 5 (here used), apply the following mathematical operations which are performed.
   a. Ascii - fifth integer
   b. Then Ascii +fourth integer*5
   c. Then Ascii + third integer * 20
   d. Then Ascii- second integer
   e. Then Ascii- first integer

2. Inserting the changed value of code in array

3. Inserting the decrypted code value

4. The random string generated is shown by considering and processing the four parameters taken.

5. From the random string generated, the particular character is retrieve

6. The outputted text is the original message.

3. DIGITAL STEGANOGRAPHY

Digital steganography is the practice of hiding the data in the digital images which will be unnoticeable to others [6, 10]. A digital image is "an array of numbers that represent light intensities at various points". These light intensities or pixels are combines to form the image's raster data. i.e digital image is a grid of squares, each of which contains a single color each square is called a pixel (for *picture element*) The images can be of 8-bits or 24-bits.
A typical digital steganographic encoder consist of the *message* which is the data that the sender wishes to send confidentially to the receiver, the *cover which* is the medium in which the message is embedded and serves to hide the presence of the message. The image with the secretly embedded message produced by the encoder is the *stego-image*. In addition, the encoder usually employs a *stego-key* which ensures that only recipients who know the corresponding decoding key will be able to extract the message from a stego-image. A typical steganography mechanism is shown in figure1.

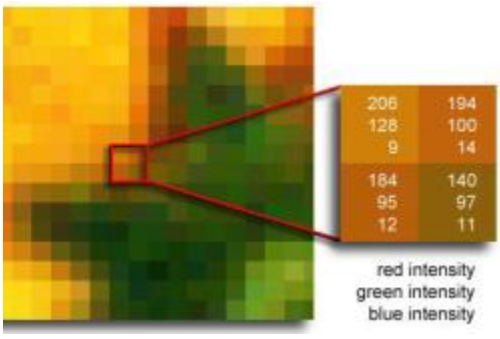

Figure 3: An image visions

Following are the main steps defined for text hidden and retrieval in/from image:

3.1 Hiding Information Process

Step 1 – Input the target image with any format bmp, gif, jpeg, and tiff.

Step 2 – Input the target text file to be hidden in the image.

Step 3 – Hide the target text in the target image using LSB technique describes in section 4.

3.2 Retrieving Information Process

Step 1 – Input the target image containing hidden text file

Step 2 - Retrieve the hidden text back from the image

Step 3 – Output is the retrieved text.

4. LSB (Least Significant Bit)Technique

This technique has been introduced in several papers [7, 8, 9, 10]. So staying on the subject, here present the procedure which is carried out for embedding the text using the LSB of pixels values. A 1,024 X 768 image has the potential to hide a total of 2,359,296 bits (294,912 bytes) of information. . Each pixel value contains the value of the color and is represented in bits (0 & 1).Similarly, text is also represented in bits (0&1).Therefore comparing bit values byte by byte result in hiding the bit values of Text in bit value of an Image. i.e. storing in LSB of a byte (pixel). For example, the letter A can be hidden in three pixels (assuming no compression). The original raster data for 3 pixels (9 bytes) may be:
(00100111 11101001 11001000)
(00100111 11001000 11101001)
(11001000 00100111 11101001)

The binary value for A is 10000011. Inserting the binary value for A in the three pixels would result in

(00100111 11101000 11001000)
(00100110 11001000 11101000)

(11001000 00100111 11101001)

The bold color bits are the only three actually changed in the 8 bytes used. On average, LSB requires that only half the bits in an image be changed. So you can hide data in the least and second least significant bits and still the human eye would not be able to discern it.

5. PSNR (Peak Signal to Noise Ratio):

For measuring the quality of reconstructed image as compared to the original image, the metric needs to be define [3, 4]. There are three common error metrics used for estimating noise on images are RMSE, PSNR, and SSIM. The PSNR is defined as:

PSNR = 10 x log10 x (MAX / RMSE)

, where, MAX is the maximum pixel value of the image. In the case of 8 bits gray scale images the MAX value will be 255.

The RMSE (Root Mean Square Error) can be defined as follows:

Error = $\sqrt{[\frac{1}{MN}\sum_{i=0}^{M-1}\sum_{j=0}^{N-1}(s-r)^2]}$

Where i, j are the positions of pixels in the image, si,j refers to the ith row and jth column pixel value of the source image and ri,j refers to the ith row and jth column pixel value of the reproduced image.

Assuming pixel values in the range [0,255], The following observations are mentioned as:

- An RMSE of 25.5 result in a PSNR of 20 and that of 2.55 results in a PSNR of 40.
- RMSE of zero which means an identical image results in infinite or undefined PSNR
- RMSE of 255 result in PSNR of zero
- RMSE greater than 255 results in negative PSNR.

As PSNR and RMSE failed in case of Gaussian noise so another metric has been introduced known as SSIM (Symmetrical Structured Index Metric) which takes three properties luminance, contrast, and structure measurement of the image.

6. IMPLEMENTATION

A. The first part which is a encryption /decryption process is implemented in C#.NET and the output of encryption is save in a file which is needed at the time of hiding the content in an image. The screen-shot is taken and the result is shown as below:

Figure 4: Program Code Snapshot

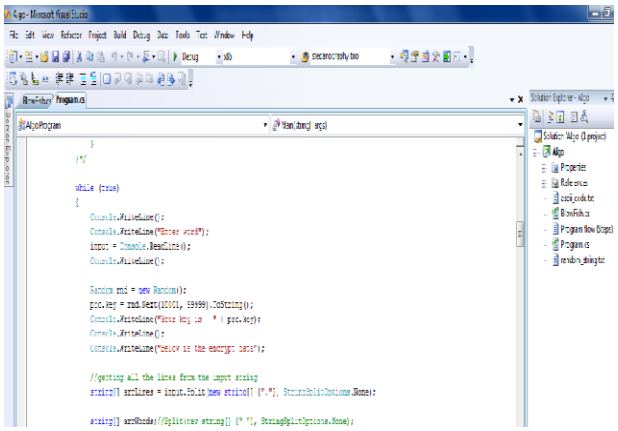

Figure 5: Output of Program

B .In the next phase, the encrypted file is taken as input file and the target image is loaded form the system. The file is hiding in the image and this implementation is done in C#.NET and the screen shot is taken. The result is shown below:

Figure 6: Information Hiding Process

The retrieval of data file is done as shown:

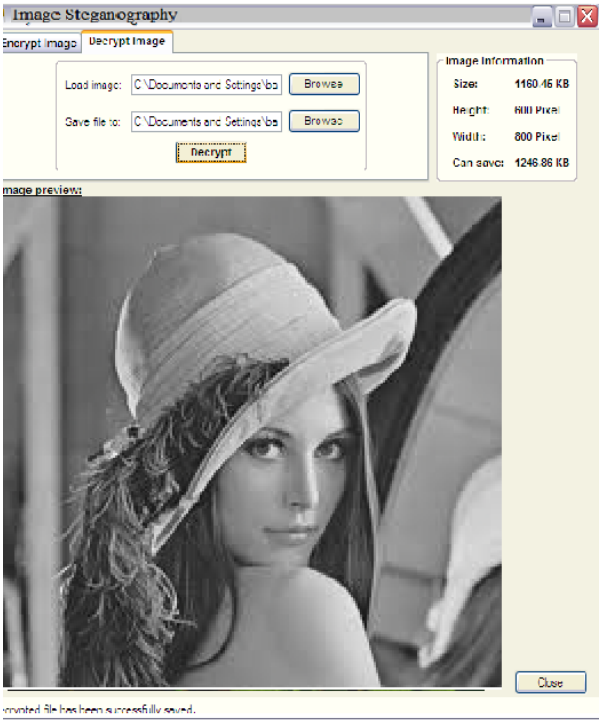

Figure 7: Retrieval Process.

C. The last phase is the PSNR calculation and analysis which is implemented in MATLAB [11]. The function code is written in Editor window for PSNR calculation which is executed and the PSNR for the two images – the input image and the target image (containing hidden text) is calculated which comes out to be 99.4191. The absolute difference is also calculated between the images which come out to be 0 (zero), which shows an identical images with quality that the effects are unnoticeable to human eyes.

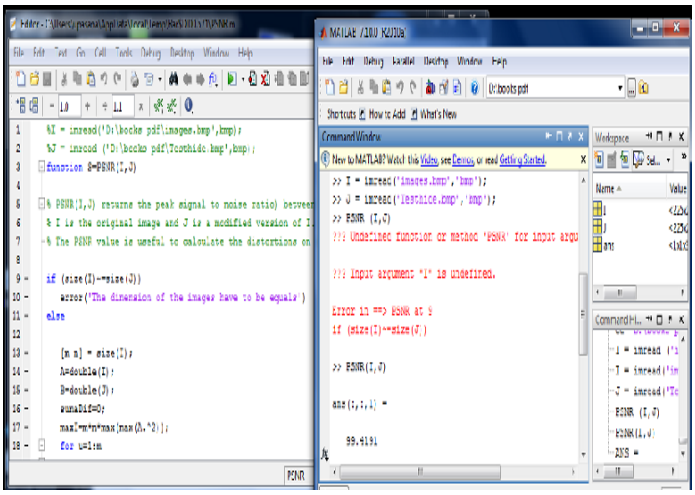

Figure 8: MATLAB Implementation Of PSNR calculation

The values have been calculated using PSNR and RMSE formulae and the graph is plot as shown:

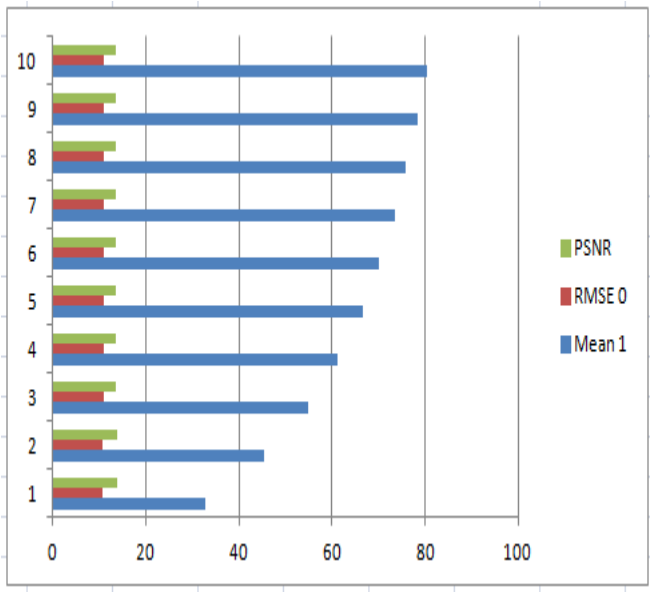

Figure 9: Bar graph represent the correspondence values of PSNR and RMSE

7. CONCLUSION

The data before sending is encrypted and then some mathematical operation is applied which add a new layer

of security. Further operations on ASCII codes can be formulated to get strong encryption. The PSNR of two images is calculated through MATLAB which is a nominal value which means that the image quality is maintained to a level. The values calculated for the image is depicted by a bar graph which shows the RMSE and PSNR. Further metric can be used to measure the quality of the image.

REFERENCES

[1] Ajit singh and Rimple Gilhotra " Data Security Using Private Key Encryption system Based On Arithmetic Coding" International Journal of Network Security & Its Applications (IJNSA), Vol.3, No.3, May 2011

[2] Tarun Narayan Shankar and G. Sahoo "Cryptography by Karatsuba Multiplier with ASCII Codes" 2010 International Journal of Computer Applications (0975 – 8887) Volume 1 – No. 12

[3] Le Phu Dung, Srinivasan Bala, Mohammed Salahadin, Kulkarni Santosh and Wilson Campbell, "A Measure for Image Quality", ACM, pp. 513-519, 1998.

[4] Z. Wang and A. C. Bovik, A universal image quality index, IEEE Signal Processing Letters, vol. 9, no. 3, pp.81-94, March 2002

[5] R. Anderson and F. Petitcolas, "On the limits of steganography," IEEE Journal on Special Areas in Communications, Vol. 16, No. 4, pp. 463-473, May 1998.

.

[6] N. Provos, and P. Honeyman, "Hide and Seek: An Introduction to Steganography," IEEE Security and Privacy, 1(3): 32-44,May 2003.

[7] C. Kim, E.-J. Yoon, Y.-S. Hong, and H. 1. Kim, "Secret Sharing Scheme Using Gray Code based on Steganography," Journal of the Institute of Electronics Engineers of korea, 46(1): 96-102, January 2009.[3] C. C. Thien, and J. C. Lin, "Secret Image Sharing," Computers and Graphics, 26(1):765-770, February 2002.

[8] Beenish Mehboob and Rashid Aziz Faruqui "A Steganography Implementation" IEEE 2008.

[9] F. Shih, Digital watermarking and stegnography, fundamentals and techniques. UsSA: CRC Press, 2008.

[10] Neil F. Johnson, Sushil Jajodia, George Mason University, "Exploring Steganography: Seeing the Unseen", IEEE Computers, February 1998, pp. 26-34.

[11] www. mathworks.com : An official site for MATLAB.

Author1

Dr. Ajit Singh is presently working as Chairperson of School of Engineering & Sciences in BPSMV, Khanpur Kalan (Sonepat). He is also having the additional charge as a Director of University Computer Center (UGC). He posses qualifications of B.Tech, M.Tech, Ph.D. He is a member of BOG (Board of Governors) of Haryana State Counseling Society, Panchkula and also member of academic council in the University. He published approximate 20 papers in National/ International journals and conferences and holds a teaching experience of approximate 10 years. He holds the membership of Internal Quality Assurance cell, UG-BOS &PG-BOS and the NSS advisory committee. He is also an associate member of CSI & IETE. His research interests are in Network Security, Computer Architecture and Data Structure.

Author2

Ms. Upasana Jauhari has completed her B.Tech degree in Computer Science from CCS University, Meerut in year 2008 and She is pursuing M.Tech in Computer Science from Bansathali University Banasthali from June 2010. Currently she is doing Internship from B.P.S.M.V KhanpurKalan, Sonipat. Her research interests are in Network Security and Soft Computing.