iJRASET
International Journal For Research in Applied Science and Engineering Technology



INTERNATIONAL JOURNAL FOR RESEARCH

IN APPLIED SCIENCE & ENGINEERING TECHNOLOGY

Volume: 6 **Issue:** V **Month of publication:** May 2018

DOI: http://doi.org/10.22214/ijraset.2018.5305

www.ijraset.com

Call: 08813907089 | E-mail ID: ijraset@gmail.com

International Journal for Research in Applied Science & Engineering Technology (IJRASET)
ISSN: 2321-9653; IC Value: 45.98; SJ Impact Factor: 6.887
Volume 6 Issue V, May 2018- Available at www.ijraset.com

An Overview on LSB and LSB+HUFFMAN based Steganography

Jayanthi[1], Lakshmi.M[2], Pavithra.M[3], Prakruthi.M[4]
[1]*Senior Assistant Professor,* [2,3,4]*Students of Department of Electronics & Communication, New Horizon College of Engineering, Bangalore, Karnataka, India*

Abstract: ***Now a day's internet has became common use, with increase in use of internet providing security to information is also important. Steganography is a method to provide network security. Steganography is art of secret communication. It involves hiding information in an appropriate carrier e.g., text, image, audio, video. This paper uses two techniques for Steganography (text into image). Least Significant Bit (LSB) and Least Significant Bit with Huffman code (LSB+HUFF) and compares result using Peak Signal to Noise Ratio (PSNR).***
Keywords: LSB (Least Significant Bit), LSB+HUFFMAN, ASCII Code, MSE, PSNR.

I. INTRODUCTION

Cryptography and Steganography are two most widely used mechanisms that provide secrecy and security for communication. Cryptography achieves this by encrypting the data. Steganography is derived from Greek words, Steganos means "cover" and Graptos means "writing" which literally means "cover writing". Generally steganography means "invisible" communication i.e., hiding messages or information in another medium like audio, video, image, communication. In simple words, it is hiding information into other information. Steganography doesn't alter the structure of the secret message, but hides it inside a cover-object. After hiding, cover object and stego-object are similar. Due to invisibility or hidden factor it is difficult to recover information without known procedure in steganography. Detecting procedure of steganography is known as Steganalysis. Image steganography is a method of hiding information in a cover-image that generates a stego-image. This stego-image is then sent to the receiver by any medium, where the third party does not know that this stego image has hidden message. After receiving stego-image, hidden message can be extracted with or without stego-key which is used in embedding algorithm. In this paper, we provide an approach in which the secret message is compressed using Huffman method and inserting it in to the LSB pixels of cover image and retrieving the message using same technique. The remaining sections of this paper are organized as follows. A brief overview of the related work and literature survey is presented in section2. LSB steganography technique in section 3. The proposed technique is presented in section 4 with an example. The result and analysis of both the techniques is discussed in section 5, followed by conclusion in section 6.

II. RELATED WORKS

Several steganographic methods have been proposed and implemented that uses LSB substitution method in literature survey. LSB substitution method embeds the secret data bits into Least Significant Bits (LSBs) of cover image pixels. There are two types of LSB steganography- LSB method and the LSB+Huffman method. The LSB replacement directly replaces the LSBs of the cover images with secret message bits. On the other hand, in LSB+Huffman method, lossless data compression method is used where hidden data is compressed using Huffman coding. The LSB+Huffman method is challenging to detect than the simple LSB replacement method.

Suchi Goyal et al [2] in their paper proposed an enhanced detection of the 1-2-4 LSB steganography and RSA cryptography in Gray Scale and Color images. For color images, they apply 1-2-4 LSB on component of the RGB, then encrypt information applying RSA technique. For Gray Images, they use LSB to encrypt information and also detect edges. In the experimental outcomes, calculates PSNR and MSE. This method makes sure that the information has been encrypted before hiding it into an input image.

Manu Devi et al. [3] proposed an improved LSB based steganography technique for images imparting better information security for hiding secret information in images in which they insert data only in the least significant byte i.e. blue component of a pixel as that having lowest contribution to the color image according to human visual system investigation and other method to embed the data in smooth areas 1-3-4 LSBs insertion technique has been utilized which hides data in 1 bit in 1 LSB of Red component (Most significant byte), 3-bits in 3 LSBs of Green component and 4 bits in 4 LSBs of Blue component (least significant byte) of each

©IJRASET: All Rights are Reserved

selected pixel. Zinia Sultana et al. [5] proposed a technique which adds double layer security to hide data in image using LSB algorithm, AES-128 encryption and a new approach of choosing index of image pixel. Secondly, they developed a steganography tool using proposed technique and evaluated the performance of the proposed technique using MSE, PSNR and by payload capacity which measures how much data can be hidden in an image using the steganography tool. The drawback of proposed method is during implementation a warning is given requesting to increase the image size or decrease the secret text size, if the payload capacity ratio is not maintained and the tool designed is not updated for compressed images.

Sherin Sugathan et al [8] proposed a new algorithm for LSB replacement based image steganography for RGB color images. The directional aspects of embedding data is explored to develop an improved LSB embedding technique. The results report an improvement in image quality measured by means of PSNR and MSE. The paper reported a simple LSB replacement approach that can improve the quality of a stego image. The results indicate that the proposed method performs well especially when embedding secret data at higher LSB bit positions. The data parallel nature of the problem makes it suitable for implementation in parallel hardware. The position of the direction bit can also be shifted and the position can even act as part of a key for extracting the secret data.

III. LSB STEGANOGRAPHY

Although Least Significant Bit (LSB) embedding algorithm is one of the most common algorithm because of its features such as simple algorithm, encryption fast, easy to implement, a large amount of hidden, still occupies an important position in the field of information hiding. The easy way to secret information within the cover file is called LSB insertion. In this technique, the binary representations of the secret data have been taken and the LSB of each byte is overwritten within the image. Consider an example, supposing that there are three neighbouring pixels (nine bytes) with the following RGB encoding:

00101011	01010011	11110101
10001001	11010000	10100100
01101110	10101001	01010110

Then their values after the insertion of 'B' will be as:

00101011	0101001**0**	1111010**0**
1000100**0**	11010000	1010010**1**
01101110	10101001	01010110

(The values in bold are the ones that were modified by the transformation). Figure 3 shows an overview of LSB steganography.

A. Embedding Algorithm

Figure 1 shows a diagram for encoding message in cover image.

1) Step 1: Read the cover image and text message, which is to be hidden in the cover image.
2) Step 2: Convert pixels of a cover image to binary value.
3) Step 3: Convert text message to ASCII character.
4) Step 4: Convert ASCII value of message to binary value.
5) Step 5: Replace LSB of the cover image with each bit of secret message one by one.
6) Step 6: Display a stego image.

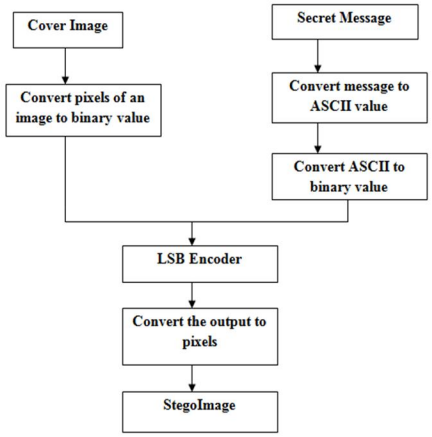


Figure 1: Diagram for embedding algorithm

©IJRASET: All Rights are Reserved

B. Algorithm to Retrieve Text Message:

Figure 2 shows a diagram for retrieving text message

1) Step 1: Read the stego image.
2) Step 2: Calculate LSB of each pixel of stego image.
3) Step 3: Retrieve bits and convert each 8 bit into equivalent ASCII character.
4) Step 4: Convert ASCII value to message string and we get secret message.

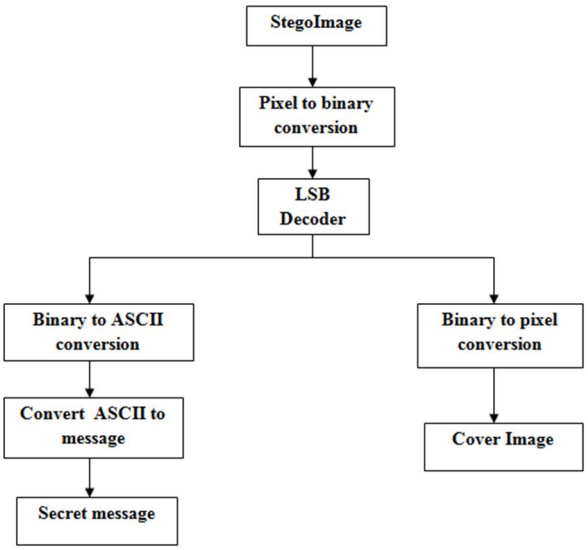


Figure 2: Diagram to retrieve the text message

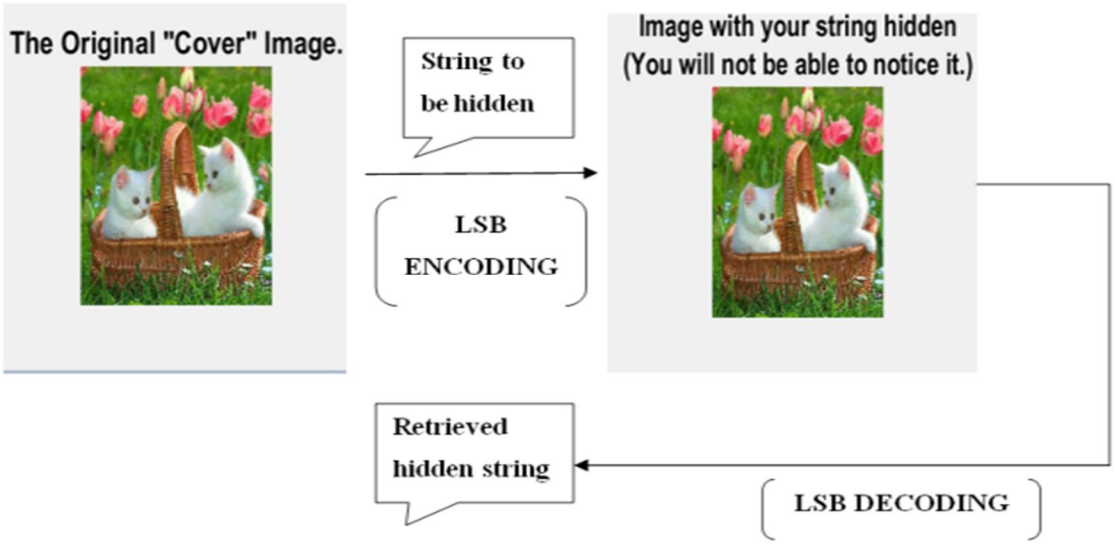


Figure 3: An Overview of LSB Steganography

IV. PROPOSED METHOD

Huffman coding is data compression techniques invented by David Huffman. It is optimal prefix code generated from set of probabilities and has been used in various compression applications. These codes are of variable code length using integral number of bits. This idea causes a reduction in the average code length and thus overall size of compressed data is smaller than the original.

Huffman encoding is a lossless data compression algorithm which means there is no loss of data in compression nor while retrieving the original data .In this algorithm, the character which occurs the most, gets the smallest code. It is used to efficiently encode characters into bits. Huffman coding tree is built in order to calculate these codes. Huffman coding tree is a binary tree.

©IJRASET: All Rights are Reserved

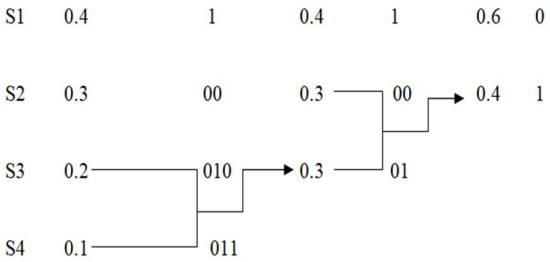
From a text, character and their occurrences are calculated and are stored in increasing order. First two nodes, i.e., characters with lowest frequency are removed from the list and combined in one node and their frequencies are also added. Then that node is placed in the least and list is sorted again. This process is repeated until only one node remains. This is the root of the tree and every node with a character is a leaf. The nodes that are formed in between are middle nodes. Every left path is assigned a value '0' and every path on right is assigned value "1". Therefore it is a tree based encoding in which one starts at the root of the tree and searches the path till it end up with the leaf. The value at the leaf is the character represented by the encoded bits. An overview of LSB+Huffman method is shown in Figure 8.

C. Procedure to Construct Binary Tree Bits

1) Source symbols are arranged in the decreasing order of their probabilities.
2) The last 2 symbols are combined into a single composite symbol by adding their probabilities. These reduced sources are designated as 'Sa'.
3) The last 2 symbol of this reduced source 'Sa' are further combined into single symbol in the same way to get further reduced source 'Sb'.
4) This process is further continued until we have only 2 symbol '0' and '1' respectively. ('0' for higher probable and '1' for lower probable)
5) Now the encoding is continued backward with respect to '0' and '1' until the completion of the encoding process up to the first column.

D. Example

Consider the symbols S1, S2, S3 and S4 with the respective probabilities of 0.4, 0.3, 0.2 and 0.1 as shown in Figure 4.



Symbol	Binary
S1	1
S2	00
S3	010
S4	011

Figure 4: An example of Huffman encoding

We'll use Huffman's algorithm to construct a tree that is used for data compression as shown in Figure 5

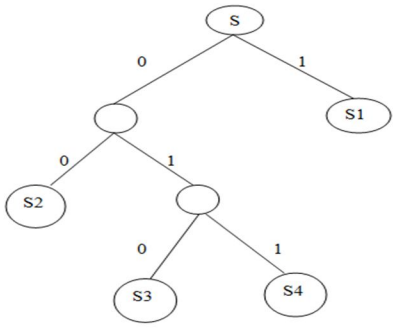


Figure 5: Huffman tree for data compression

©IJRASET: All Rights are Reserved

Huffman code is an algorithm to compression based on the frequency of occurrence of a symbol in the text message.

E. Embedding algorithm for proposed method:

Figure 6 shows a diagram for encoding message in cover image using LSB+Huffman method.

1) Step 1: Read the cover image and text message, which is to be hidden in the cover image.
2) Step 2: Convert pixels of a cover image to binary value.
3) Step 3: Compress the string to be hidden using Huffman encoding.
4) Step 4: Convert Huffman encoded text message to ASCII character.
5) Step 5: Convert ASCII value of Huffman encoded message to binary value.
6) Step 6: Replace LSB of the cover image with each bit of Huffman encoded secret message one by one.
7) Step 7: Display a stego image.

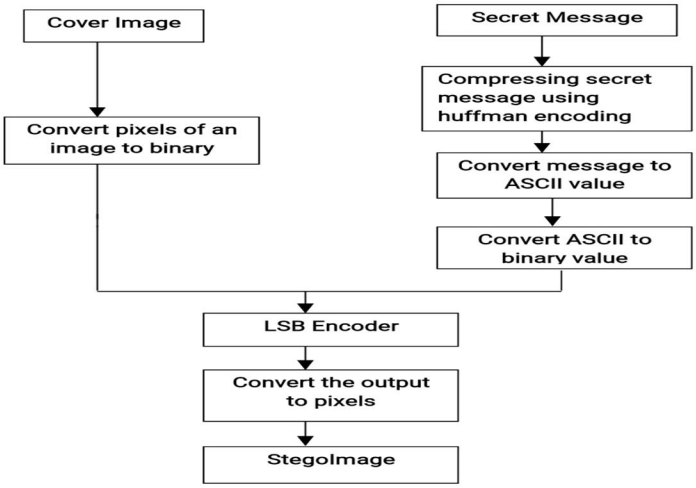


Figure 6: Diagram for embedding algorithm using proposed method

F. Algorithm for Retrieving text Message for Proposed Method

Figure 7 shows a diagram for retrieving message using proposed method.

1) Step 1: Read the stego image.
2) Step 2: Calculate LSB of each pixel of stego image.
3) Step 3: Retrieve bits and convert each 8 bit into equivalent ASCII character.
4) Step 4: Convert ASCII value to message string and we get Huffman encoded message.
5) Step5: Decompress the Huffman encoded message using Huffman decoding algorithm to get back hidden secret message.

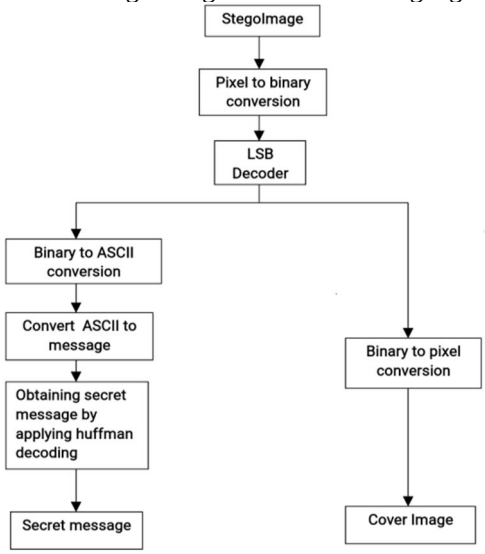


Figure 7: Diagram to retrieve the text message using proposed method

©IJRASET: All Rights are Reserved

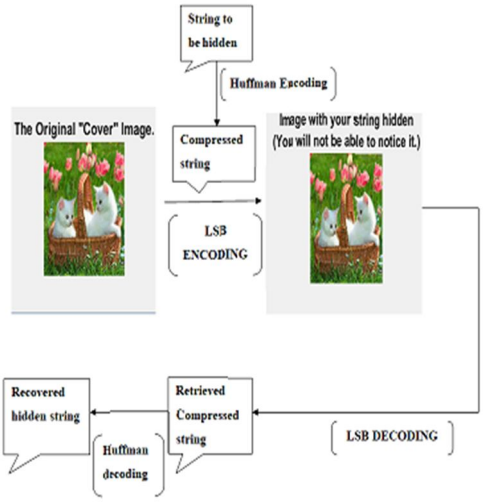


Figure 8: An Overview of LSB+Huffman steganography method

V. EXPERIMENTAL RESULT AND DISCUSSION

A. Experimental Results

The proposed method was tested on different RGB color images. For the analysis of experiment, different random alphanumeric messages were generated for every images. The same message is embedded into a cover image using two schemes for comparing the quality of an image. The secret message was embedded using both the Algorithm in the lowest Bit plane and the comparison results obtained for different cover images are represented in the form of image quality factor known as The Mean Squared Error (MSE) and Peak Signal to Noise Ratio (PSNR) values obtained are shown in the following Table1

1) *Mean Square Error (MSE):* It defines the square of error between original image and stego image

$$\text{MSE}=10\log_{10}\sum_{n=0}^{N}[x(n)-y(n)]^2$$

Here x(n) represents cover image and y(n) represents stego image.

2) *Peak Signal to Noise Ratio (PSNR):* It measures the quality of image. PSNR compare the original image with stego image. PSNR is measured in decibels(db).

$$\text{PSNR}= 10\log_{10}\frac{255^2}{MSE}$$

Comparison between the techniques used we consider all images of almost same size and string hidden is "steganography is art of secret communication" When all the images taken are of almost same size then there won't be much difference in the PSNR and MSE values. The comparison result for both the technique are tabulated which shows image quality factor like PSNR and MSE values in Table 1

IMAGE	SIZE (KB)	LSB STEGANOGRAPHY		LSB+HUFFMAN STEGANOGRAPHY	
		MSE	PSNR	MSE	PSNR
	9.59	0.00113	77.6303	0.00048	81.268
	6.88	0.001136	77.6094	0.0005	81.099
	4.77	0.001233	77.2539	0.00054	80.827

Table 1.PSNR and MSE value for the user provided string

©IJRASET: All Rights are Reserved

B. Discussion

In this work, we review two different Steganography algorithms for a hidden message inside a cover-image. LSB and LSB + Huffman.

Further, the most commonly used method that is the Least Significant Bit (LSB) method is discussed and a description is given as to how image steganography can be implemented using this method by embedding the binary values of each character in the secret text. Then we proceed with LSB+Huffman coding where Huffman encoding and decoding algorithm is used to compress and decompress the hidden string. The use of Least Significant Bit + Huffman Coding is shown and its comparison with the former mentioned method is done. The Least Significant Bit (LSB) method is used to embed the secret data by directly using the binary values of each character in it, the total number of bits used is significantly more as compared to the bits required when using LSB with Huffman Coding. Since the working of Huffman Coding algorithm revolves around the theory that frequent occurrences are given less bits and less occurrence are given more bits, variable length codes assigned to each character in the secret data cuts down the total bit requirement by a significant number. Output of, LSB and LSB+Huffman is compared using MSE and PSNR ratio to find which technique is best. A conclusion can be made from the results shown above that LSB+Huffman provide greater PSNR and is better technique.

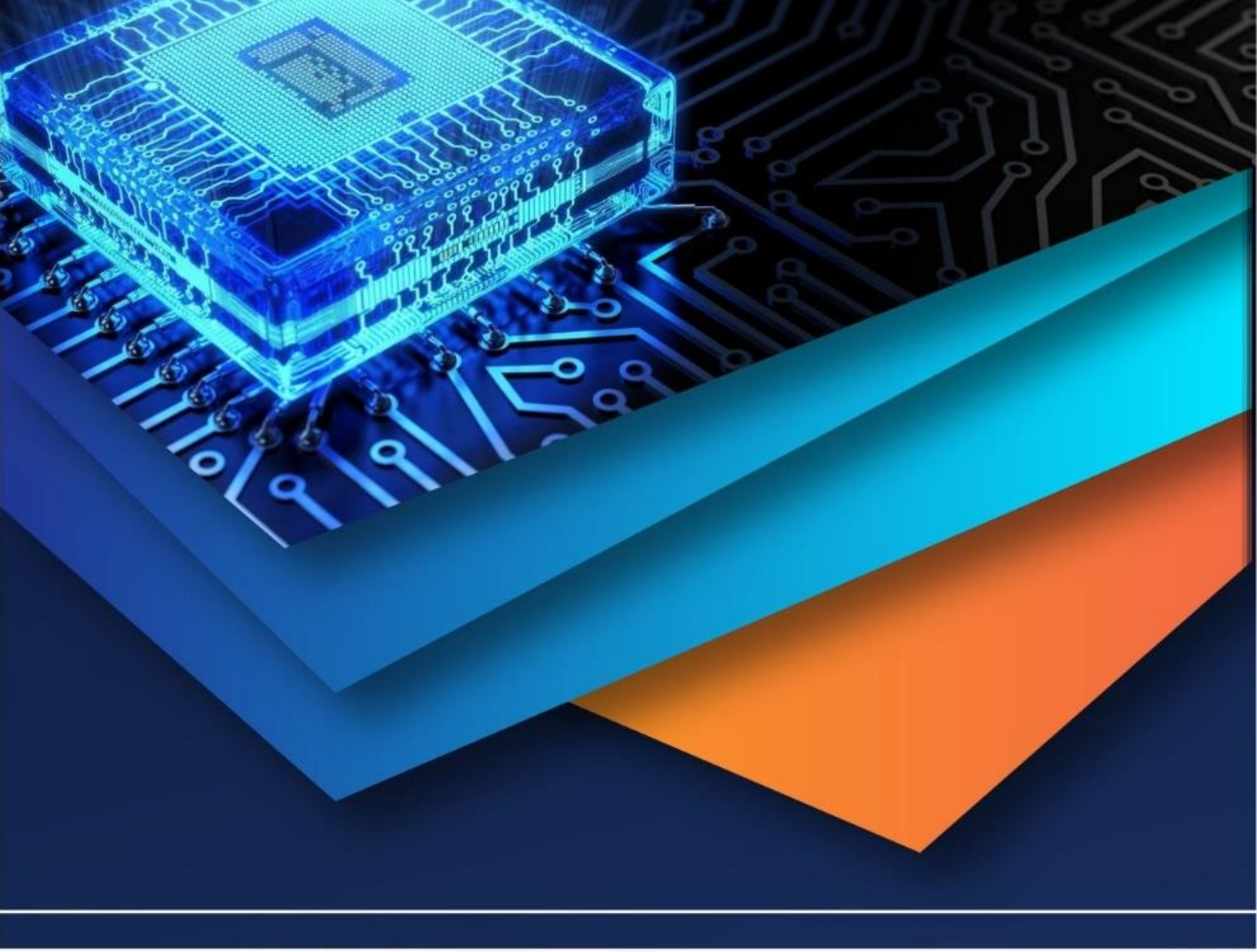
VI. CONCLUSION

In this paper we have gone through LSB and LSB+Huffman code based image data hiding technique (image steganography). By comparing the results of the two techniques we conclude that LSB+Huffman method is better than LSB method to hide text into image because LSB+Huffman method is challenging to detect than the simple LSB insertion steganography method and also gives greater PSNR value when compared to that of LSB method since we hide the compressed string (secret message is compressed using Huffman coding technique) not the secret message as in LSB technique.

REFERENCES

[1] Kavitha, Kavita Kadam, Ashwini Koshti, PriyaDunghav “Steganography Using Least Significant Bit Algorithm”, International Journal of Engineering Research and applications, vol.2, issue 3, pp. 338-341 May-June2012
[2] Suchi Goyal, Manoj Ramaiya, Deepika Dubey ”Improved Detection of 1-2-4 LSB Steganography and RSA cryptography in color and Grayscale Images”, International Conference on Computational Intelligence and Communication Networks,201
[3] Manu Devi, Nidhi Sharma ” Improved Detection of Least Significant Bit Steganography Algorithms in Color and Gray Scale Images”, RAECS UIET Punjab University Chandigarh,06-08 March 2014.© 2014 IEE
[4] J. Hossain, “Information-Hiding Using Image Steganography with Pseudorandom Permutation”, Bangladesh Research Publications Journal, vol.9, no. 3, pp. 215-225, 2014
[5] Zinia Sultana , Fatima Jannat , Muhammad Nazrul Islam ” A New Approach to Hide Data in Color Image Using LSB Steganography Technique”,3rd International Conference on Electrical Information and Communication Technology (EICT), 7-9 December 2017, Khulna, Bangladesh.
[6] Arvind K. and Kim P “Steganography- A Data Hiding Technique”, International Journal of Computer Applications, Volume 9 , November 2010.
[7] Fridrich, J., Goljan, M., "Reliable detection of LSB steganography in color and grayscale images", Proc. ACM Workshop on Multimedia and Security (2001)27–30
[8] Sherin Sugathan “An Improved LSB Embedding Technique for Image Steganography”, 2016 IEEE
[9] Gandharba Swain “A steganographic method combining LSB substitution and PVD in a block” ,International Conference on Computational Modelling and Security (CMS 2016
[10] Sreekutty S Kumar, Sylish S V ” Image Steganography in High Entropy Regions using a Key & Modified LSB for Improved Security” in Proceedings of the IEEE 2017 International Conference on Computing Methodologies and Communication (ICCMC©2017 IEEE

©IJRASET: All Rights are Reserved



10.22214/IJRASET

45.98

IMPACT FACTOR: 7.129

IMPACT FACTOR: 7.429

INTERNATIONAL JOURNAL FOR RESEARCH

IN APPLIED SCIENCE & ENGINEERING TECHNOLOGY

Call : 08813907089 (24*7 Support on Whatsapp)