Data Compression and Steganography Using Shift LSB Algorithm

Himadri Parikh[1], Jay Amin[2]

[1] PG Student, Information Technology, LJIET, Ahmedabad, Gujarat, India
[2] Assistant Professor, Information Technology, LJIET, Ahmedabad, Gujarat, India

ABSTRACT

Steganography is the method of writing hidden or secret messages in such a way that no attacker can suspect the existence of the secret message. Only the receiver can know that there exits some message related to him. Since Steganography is the Greek word, it means "Covered Writing". Using this, one can transmit secret data in an insecure channel as it hides the message inside another message which can be text, audio, video or image. In this research work, in order to increase more security, along with steganography, data compression is used. Initially the secret data is being compressed using Huffman coding technique using the compression tools. Then the compressed data is stored in an image using Shift LSB (Least Significant Bit) Steganography technique. Since the message is compressed first, it reduces the size of file that is to be transmitted. Thus it increases security as well as the bandwidth is consumed less. And also the speed of transmission is faster and easier as the file used here of small size which contains the compressed image.

Keyword: - *Huffman coding, LSB Algorithm, Data Compression, Steganography, PSNR, MSE*

1. INTRODUCTION

Information hiding or providing stake to the confidential messages sent over the transmission medium which is dangerous, it can be provided via cryptography, steganography and watermarking. Nowadays in this IT industry preferably importance is given on how to secure data of any organization. Steganography acts as the dissimulation to the fact that the communication has occurred, by concealing information in other information. It conceals the secret message in such a way that the pristine message appears to be same though it contains the secret message within itself. It is derived from a Greek word; it signifies hidden or secret writing. Thus the goal here is always to conceal or cover the very existing of the embedded data. Today, steganography is most often associated with data hidden with other data in an electronic file. This is conventionally done by superseding that least consequential or most redundant bits of data in the pristine file.

Where CRYPTOGRAPHY scrambles a message into a code to obscure its construal, steganography conceals the message entirely.

Steganography is the way that involves communicating secret data in a felicitous multimedia carrier, e.g., image, audio and video files. The media with or without concealed information are called Stego Media and Cover Media, respectively. Steganography can meet both licit and illicit fascinates, e.g., civilians may utilize it for preserving privacy while terrorists may utilize it for spreading terroristic information.

Two other technologies that are kindred to steganography are watermarking and fingerprinting. These technologies are mainly concerned with the aegis of perspicacious property. But steganography is concern with the concealing of text in information like image, text, audio, and video.

This paper presents a technique for steganography using Modified LSB based on Huffman encoding. In the proposed system, initially the text is compressed using the Huffman encoding. The result from it is the input for the LSB algorithm where the bits are first embedded into the 32-bits and then the left shift is performed on those bits so that it provides more security.

1.1 Steganography

Steganography is the art of overlaying the truth that communication exchange is taking place, because it conceals the knowledge in other information. It's the art of hiding a message in a cover without leaving an amazing track on the pristine message. The intention of Steganography is to duvet the very presence of communication making the true message not recognizable to the observer. In steganography the information is stored secret without any changes but in cryptography the long-established content material of the message is differed in one-of-a-kind stages like encryption and decryption.

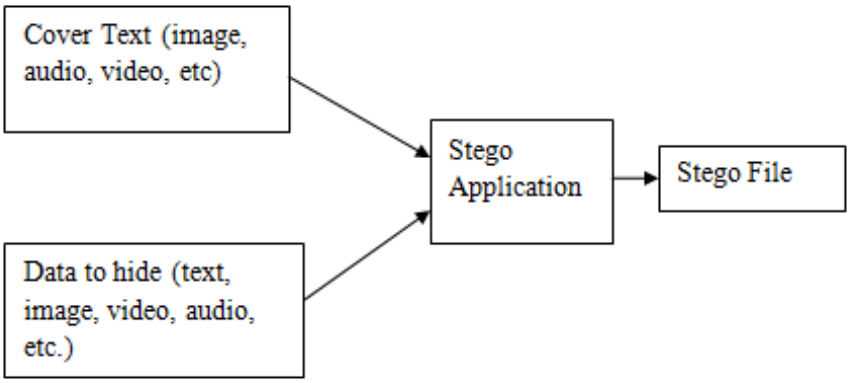

Fig. 1: Steganography

Types of Steganography

- Text
- Image: Image steganography is widely use for hiding process of data. Because this is quite simple and secure way to transfer the information over the internet. Image steganography has following types:
  * Transform domain 1) Jpeg 2) Spread Spectrum 3) Patch Work
  * Image domain 1) LSB and MSB in BMP 2) LSB and MSB in JPG
- Audio
- Video
- Protocol

1.2 Data Compression

Data compression is a procedure through which a file (text, Audio, and Video) could also be modified to one more (compressed) file, such that the normal file could also be completely recovered from the long-established file without any loss of exact knowledge. This process may be subsidiary if one wishes to save lots of the storage space. For instance if one wishes to retailer a 4MB file, it is usually top-rated to first compress it to a smaller size to save the storage space. Additionally compressed files are much more effectively exchanged over the web for the reason that they add and down load much faster. We require the potential to reconstitute the original file from the compressed variation at any time. Data compression is a method of encoding rules that sanctions substantial reduction in the total number of bits to store or transmit a file. The more data being handled, the more it costs in phrases of storage and transmission costs. In short, Data Compression is the method of encoding data to fewer bits than the customary illustration in order that it takes less storage space and not more transmission time even as communicating over a network.

There are two mainly two types of Data Compression: [8]

1. Lossy Compression
2. Lossless Compression

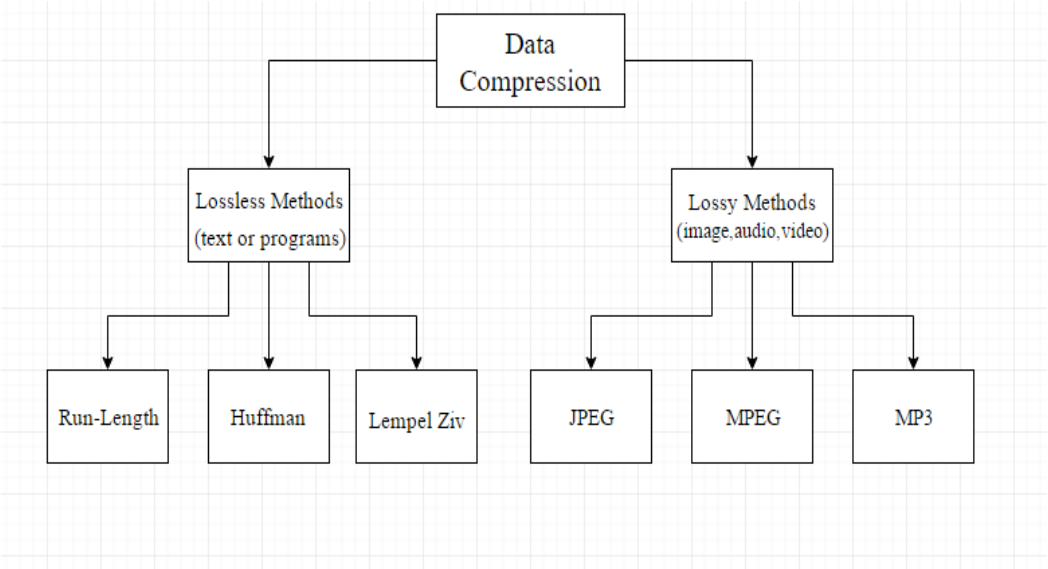

Fig. 2: Types of Data Compression

1.3 Huffman Coding

A frequency predicated coding scheme (algorithm) that follows Huffman's conception is called Huffman coding. Huffman coding is a simple algorithm that engenders a set of variable-size code words of the minimum average length.

The algorithm for Huffman encoding includes the following steps:

1. Construct the frequency table sorted in descending order.
2. Building a binary tree: Carrying out iterations until completion of a complete binary tree:
   - Merge the last two items (which have the minimum frequencies) of the frequency table to form a new combined item with a sum frequency of the two.
   - Insert the combined item and update the frequency table.
3. Deriving Huffman tree: Starting at the root, trace down to every leaf; mark '0' for a left branch and '1' for a right branch.
4. Generating Huffman code: Collecting the 0s and 1s for each path from the root to a leaf and assigning a 0-1 codeword for each symbol.

Example: S0 = {A, B,....., E} and p(A) = 0.1 = p(B), p(C) = 0.3, p(D) = p(E) = 0.25. The nodes in S are shown shaded.

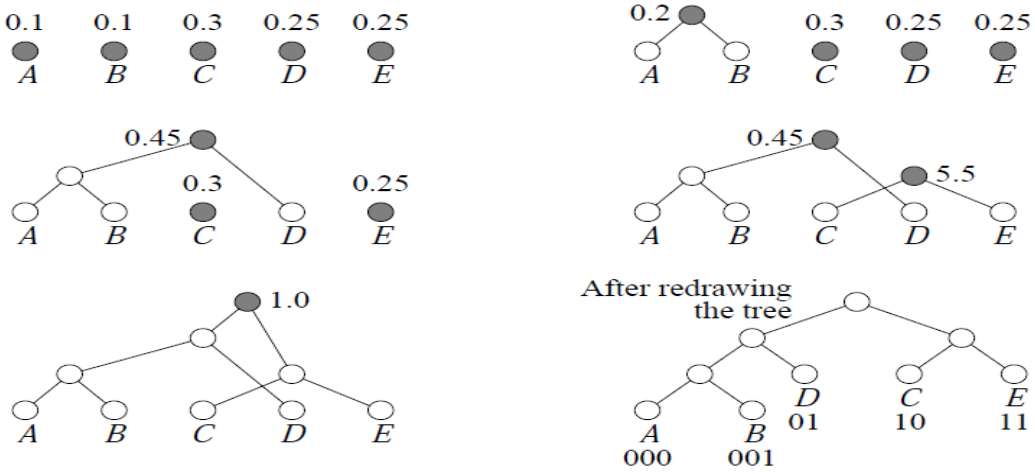

Fig. 3: Huffman Coding[9]

1.4 Least Significant Bit

The least significant bit insertion method is probably the most prominent image steganography technique. It is a prevalent, simple approach to embedding information in a graphical image file. Lamentably, it is prodigiously vulnerably susceptible to attacks, such as image manipulation. A simple conversion from a GIF or BMP format to a lossy compression format such as JPEG can eradicate the concealed information in the image. When applying LSB techniques to each byte of a 24-bit image, three bits can be encoded into each pixel. (As each pixel is represented by three bytes) Any transmutations in the pixel bits will be indiscernible to the human ocular perceiver.

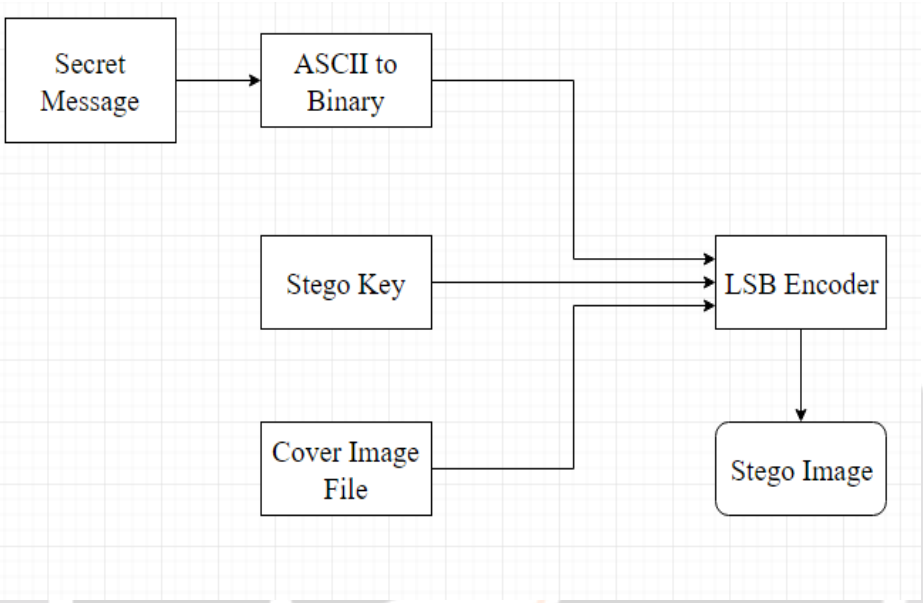

Fig. 4: LSB Algorithm

For example, the letter A can be hidden in three pixels. Assume the original three pixels are represented by the three 24-bit words below:
(00100111 11101001 11001000)
(00100111 11001000 11101001)
(11001000 00100111 11101001)

The binary value for the letter A is (10000011). Inserting the binary value of A into the three pixels, starting from the top left byte, would result in:
(00100111 11101000 11001000)
(00100110 11001000 11101000)
(11001000 00100111 11101001)

2. EXISTING WORK

The proposed algorithm includes character reversal, utilization of a substitution cipher, Huffman coding and Quadrant predicated LSB embedding.

The data embedded utilizing this algorithm does not concentrate on the categorical portion of the image but is greatly dispersed and thus increases the security.

Proposed System Algorithm:

Embedding Algorithm

Input: secret message.

Output: A stego-image.

Step-l: Splitting of image: The cover image is split up into four quadrants Q- l, Q-2, Q-3, and Q-4.

Step-2: Initial encipherment: The secret data, S which is to be processed is first reversed and a substitution cipher is carried out by adding 1 with the alphabet and a new secret information, S' is generated.

Step-3: Huffman coding: The obtained information S' is then passed to the Huffman encoder which converts them into Huffman codes. The result of Huffman coding is the sequence of bits comprising of 0 and 1.
Step-4: Key generation: The key is the 4 digit sequence that represents the order of quadrant in which the bits are to be embedded. It is generated by the pseudorandom generator and it comprises of values 1, 2, 3 and 4 occurring in a random order but exactly once. The first bit is embedded in the quadrant which is specified by the first digit of the key and second bit in quadrant specified by the second digit and so on.
Step-5: Embedding: Embedding is carried out using the LSB embedding technique. LSB embedding is performed by using the sequence of bits produced by Huffman coding generated in the step 2. Each bit is embedded in the quadrant specified by the key. The stego image is obtained after all bits of the bit sequence have been embedded.

3. PROPOSED WORK

So in the proposed system, we are going to modify LSB algorithm by using left shift in the bit pattern. But before that we are going to compress the secret data using the Huffman coding. This compressed data is then divided into the 8 bits chunks. Then these 8 bits data is padded with 24 bits so that to convert into 32 bit. This padding is used so that the attacker doesn't get know to that what exactly the size of actual message is. But the last 8 bits of the message can be of any size.
Example:
11100000111100101010101010110
This is our message. It contains 27 bits. So first divide the message into 8 bits chunks.
11100000
11110010
10101010
110
Now do the 1st three chunks are padded with 24 extra bits. And the last chunk is padded with 29 extra bits. These padded bits are chosen in random. Before doing padding in the last chunk, initially it counted that how many bits are there. Here there are only 3 bits, so it is converted into decimal. So the decimal is how many digits not the 8421 format. Thus it determines how many times we are going to do the left shifts for the above padded bits. Thus these padded bits are stored in an array.
Final[0]= 11100000101101010011100011101100
Final[1]= 11110010011011000111100011100101
Final[2]= 10101010101010001110111100000111
Final[3]= 00000110111100000110110101000101

These 32 bits chunks are left shifted by 3 times as the key is 3 digits. And thus the after left shifting every chunk, they are concatenated and thus we get the encrypted bits which are embedded in an image using the LSB algorithm.
For decoding the bits from an image, LSB is used. We get the encrypted bits. Then divide the bits into the 32 bits chunks, and then do the reverse shifts but before that we have to determine the key. The key is determined from the last 32 bits of the message. The last 24 bits are trimmed. So that we get the 8 bits. And since the key that we obtain is in the binary. We need to convert again to the decimal, which determines how many times we have to do the reverse shifts. Thus after getting 32 bits of message and by reverse shifting, we do the trim again on each 32 bits chunk. Thus we get the 8 bits chunk, and then the concatenations of those 8 bits are done. Finally we get the decoded message, but it is in the binary compressed form. The last step is to apply decompression on those bits to get the actual message.
Example: A secret message "hello" is supposed to be sent to the respective receiver.
Initially it is compressed using Huffman coding.
Thus the compressed data for the "hello" message is 1111100010.
Now the bit representation of each bit is padded with extra bits so that the size of each bit becomes 32 bits. And this 32 bits representation is then left shifted and then using LSB algorithm, the bits are stored at the last bit of every pixel bit in the cover image.

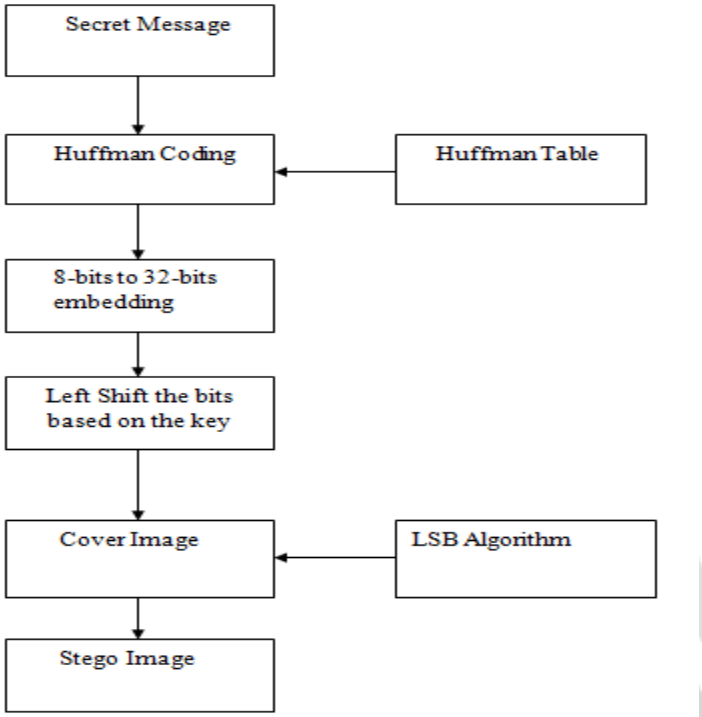

Fig. 5: Flow Chart of Proposed System

Step 1: Huffman coding.

The secret data is compressed initially using Huffman coding compression technique. In this we are going to use any compression tool like JavaScript Huffman Encoder.

Step 2: 8-bits to 32-bits embedding

Now the bit representation of each bit is padded with extra bits so that the size of each bit becomes 32 bits.

Step 3: LSB with Left Shift

These 32 bits representation for each character is then left shifted multiple times based on the key and then using LSB algorithm to embed bits into the cover image. The LSB technique directly embeds the secret data within the pixels of the cover image.

3.1 Implementation Results

In Simulation experiment, the base paper and the proposed paper are being tested on JAVA platform with few images and different text files. Here the inputs provided to the application are Cover image and the Text file. In the proposed system, the MSE and PSNR are calculated. Here the brain.jpg image is used as the cover image and xyz.txt file is used as the secret message. So after the cover image is embedded with the secret text, the Embedded Image is obtained. Thus the comparison is made here using MSE and PSNR values.

brain.jpg Image	**MSE**	**PSNR**
Existing System	0.007474441	69.39501667
Proposed System	0.0	100 (Infinity)

Table 1: Comparison using MSE and PSNR value

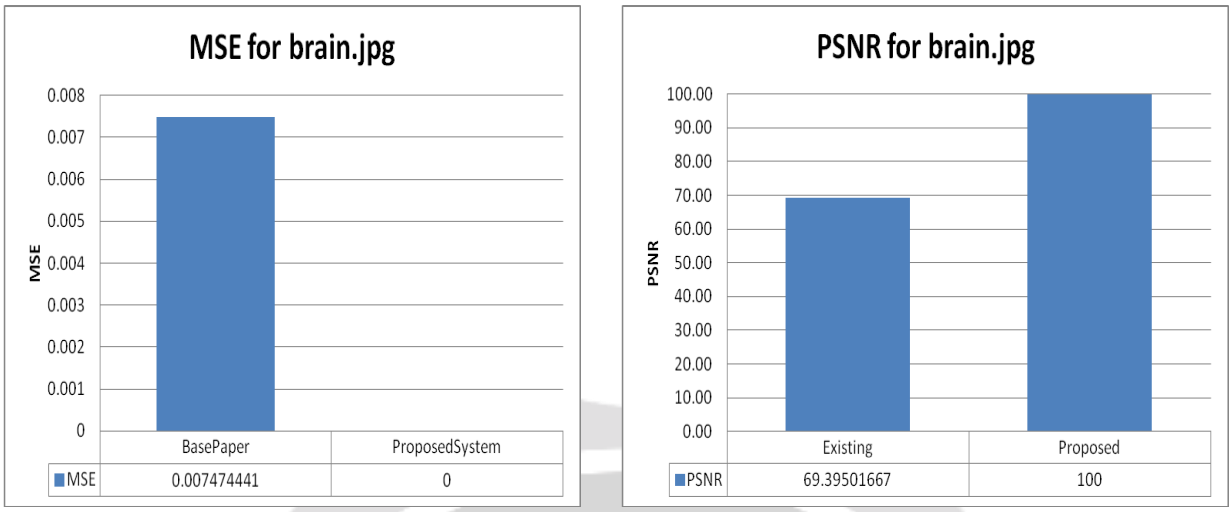

Fig. 6: Graph of MSE and PSNR between the existing and proposed system

Fig 7: brain original image (left) and stego image (right)

Correlation: Correlation is a statistical measure that indicates the extent to which two or more variables fluctuate together. A positive correlation indicates the extent to which those variables increase or decrease in parallel; a negative correlation indicates the extent to which one variable increases as the other decreases.

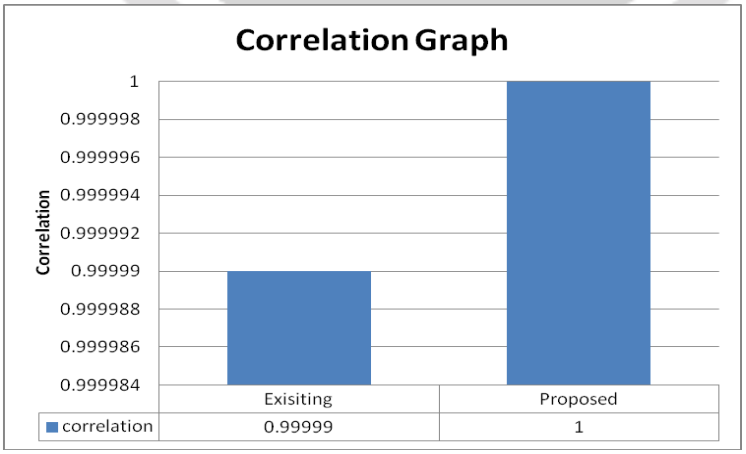

Fig. 8: Correlation graph for both the systems.

4. CONCLUSIONS

Security is an important issue while transferring the data using internet because any unauthorized individual can hack the data and make it useless or obtain information un- intended to him. Thus here initially the data which is to be hidden is compressed and then it is covered using shift LSB algorithm. Then the data is hidden in the image. This is the secure and most efficient transmission of the message. Also the secret data is being compressed first. Due to which it becomes easier to store data into the cover image as the size of the data decreases. And also the image required is of smaller size as the secret data to be stored is also less. Thus this requires less bandwidth to transfer image in the insecure channel. Also from the correlation we can determine that the stego image of the proposed system is more robust as compared to that of the existing work.

5. ACKNOWLEDGEMENT

I am very grateful to **Dr. A. C. Suthar,** Principal of L. J. Institute of Engineering and Technology for providing facilities to achieve the desire milestone.
I also extend my thanks to Head of Department **Prof. Gayatri Pandi** for her inspiration and continuous support.
I wish to warmly thank my guide**, Prof. Jay Amin** for all his diligence, guidance, encouragement, inspiration and motivation throughout. Without his valuable advice and assistance it would not have been possible for me to attain this landmark**.** He has always been willingly present whenever I needed the slightest support from him. I would not like to miss a chance to say thank for the time that he spared for me, from his extremely busy schedule. I am very obliged to all my dear friends for their continuous livelihood and comfort in each and every phase of my life.
I would like to thank all of them whose name are not mentioned here but have played a significant role in any way to accomplish the work. Grace of the almighty God and blessings of my parents have formed the path to reach my desire goal.

6. REFERENCES

1. Venkata Keerthy S, Rhishi Kishore T K C, Karlhikeyan B, Vaithiyanathan V, Anishin Raj M M., "A Hybrid Technique for Quadrant Based Data Hiding Using Huffman Coding" in IEEE Sponsored 2nd International Conference on Innovations in Information Embedded and Communication Systems ICIIECS'15, DOI 10.1109/ICIIECS.2015.7193011, Page: 1-6. ISBN: 978-1-4799-6817-6 on 19-20 March 2015
2. Wa'el Ibrahim A. Al-Mazaydeh, "Image Steganography using LSB and LSB+Huffman Code" in International Journal of Computer Applications (0975 – 8887) Volume 99– No.5, August 2014.
3. Rig Das and Themrichon Tuithung, "A Novel Steganography Method for Image Based on Huffman Encoding" in IEEE Emerging Trends and Applications in Computer Science (NCETACS), 2012 3rd National Conference on 30-31 March 2012, DOI 10.1109/NCETACS.2012.6203290, Page: 14-18. ISBN: 978-1-4577-0749-0
4. M.Vijay and V.VigneshKumar, "Image Steganography Algorithm based on Huffman Encoding and Transform Domain Method" in 2013 Fifth International Conference on Advanced Computing (ICoAC) 2013 IEEE on 18-20 Dec. 2013, DOI 10.1109/ICoAC.2013.6922005, Page: 517 - 522 Print ISBN 978-1-4799-3447-8.
5. Made Sumarsana Adi Putra, Gelar Budiman, and Ledya Novamizanti, "Implementation of Steganography using LSB with Encrypted and Compressed Text using TEA-LZW on Android" in Computer, Control, Informatics and Its Applications (IC3INA), 2014 International Conference on 21-23 Oct. 2014, DOI 10.1109/IC3INA.2014.7042607, Page: 93 – 98. Print ISBN 978-1-4799-4577-1 in IEEE.
6. M. Gomathymeenakshi, S. Sruti, B. Karthikeyan, Meka Nayana' "An Efficient Arithmetic Coding Data Compression with Steganography" in 2013 IEEE Emerging Trends in Computing, Communication and Nanotechnology (ICE-CCN), 2013 International Conference on 25-26 March 2013, DOI 10.1109/ICE-CCN.2013.6528520, Page: 342 - 345. ISBN: 978-1-4673-5037-2
7. Tahir Ali, Amit Doegar, "A Novel Approach of LSB Based Steganography Using Parity Checker" in International Journal Volume 5, Issue 1, January 2015.
8. Amandeep Singh Sidhu [M.Tech], Er. Meenakshi Garg [M.Tech], "Research Paper on Text Data Compression Algorithm using Hybrid Approach" in IJCSMC ISSN 2557 – 7987, Vol. 3, Issue. 12, December 2014, pg.01 – 10.
9. Kshetrimayum Jenita Devi, "A Sesure Image Steganography Using LSB Technique and Pseudo Random Encoding Technique" in May 2013, IJCSMC ISSN 2677 – 7997.

Websites:

10. http://www.slideshare.net/sherifghoname/data-compression-28115182, at 6:29 am Friday December 11,2015
11. http://www.csc.lsu.edu/~kundu/dstr/4-huffman.pdf, at 9:46 am Friday, December 11,2015

Books:

12. Eric Cole, Hiding in Plain Text, Wiley Publishing,Inc. :2003
13. V. K. Pachghare, Cryptography and Information Security, Prentice-hall Of India Pvt Ltd
14. "Cryptography and Network Security" by William Stallings