# Improved Image Steganography Algorithm using Huffman Codes

Hamza Tariq Khan
Dept. of Computer Science & Engineering
Jamia Hamdard University
New Delhi

Heebah Saleem
Dept. of Computer Science & Engineering
Jamia Hamdard University
New Delhi

## ABSTRACT
Steganography promises confidentiality of data being transmitted between two parties by hiding the very existence of the data. It helps to achieve secure transmission of data over a network. This research paper provides a technical introduction of image steganography and its implementation using Least Significant Bit (LSB) technique with Huffman Coding. It focuses on the use of Huffman Coding Algorithm to minimize the amount of bits to be embedded into the carrier in order to achieve efficient image steganography.

## Keywords
Steganography, image steganography, carrier image, steganography object, steganography key, Least Significant Bit (LSB), Huffman Coding.

## 1. INTRODUCTION
Communication is the need of every living person in today's world. Several methods are used by people to communicate with others such as letters, e-mails, voice notes, video conferences, etc. Irrespective of the method used for communication, the fact that the communicating data is safely and securely transmitted to the destined party is desired by all. Cryptography and Steganography are the two most widely used mechanisms that provide secrecy and security for communication. Cryptography achieves this by encrypting the data in such a way that it cannot be easily interpreted by any person in the middle of the data's source and destination. Though steganography serves to achieve the same purpose, it differs from cryptography in a way that it effectively hides the existence of data in a communication so that no third party can detect its presence. Image steganography, a type of steganography, accomplishes this goal by embedding the data in an image and therefore hiding its existence. Today, digital data and networks are used as elements of steganography based communication.

## 2. RELATED WORKS
The author reviews and does an analysis of the different techniques that are available for steganography and also mentions the common standards and guidelines that are followed when implementing it [1]. The authors discuss different image files and how secret data can be embedded into them and also evaluates the available steganographic softwares [2]. The theory of Huffman Coding Algorithm and its working principle is described [3]. The paper provides a survey of the methods used in steganography for images in spatial representation and in JPEG format and also mentions ways to improve steganography security [4]. The authors review some of the recently adopted techniques for image steganography and also define notion of security for a steganographic system [5]. The author provides an overview of image steganography, its real world applications and its techniques and tells which technique is best suited for any particular application [6]. The paper reviews the different security and data hiding techniques available for steganography [7]. The authors describe the use of symmetric key and LSB technique for on-line hiding of information [8]. The authors give an in-depth look of how images can be used as carriers for embedding secret data in them and analyses the performance of available steganographic tools [9]. The paper generally describes the uses of steganography, how it works, which softwares are commercially available and some more closely related issues with steganography [10]. The article explains the use of LSB technique for randomly dispersing the secret data bits in an image to increase security [11]. The author discusses steganography techniques and compares them on the basis of how much data can be embedded in them [12]. The embedding of secret data in an image by the use of simple LSB technique along with worst case mean-square-error between the output stego-image and the original image is shown [13]. The authors compare the effectiveness of different steganography techniques on the basis of their mean-square-error and peak signal to noise ratio [14]. The paper focuses on how the security of the secret image can be increased without distorting the carrier image too much and also how to minimize the image hiding process time [15].

## 3. STEGANOGRAPHY
The term “Embedding” is used to describe the process used by steganography to hide the secret data in any multimedia file such as video, image, audio, etc. The word steganography gets its meaning from the Greek words “stegos” and “grafia” which mean “cover” and “writing”, respectively. The idea behind the use of steganography is to hide data that is needed to be kept secret, inside another type of data in such a way that no undesired recipient can detect its presence or can have access to it. The booming networking sector today provides new ways to implement safe and secure steganography. The vast network infrastructure serves as a high-speed medium where secret data is hidden inside multimedia contents like images, audio and video using various embedding techniques which guarantee transmission of data in a safe and secure manner.

Hosting, sending and receiving of multimedia files over the internet or any other network is a very common practice nowadays. Therefore, it is easy to embed secret data in files like these and transmit it to the destined location without it being noticed by any other person on the network. Since the multimedia content can be anything ranging from a text message to video, steganography can be divided into the following types:

- Text Steganography: This type of steganography is implemented by hiding a byte of secret data inside the nth letter of every word in the text message or

file that is being sent as a multimedia content or by manipulating the text in the file in ways such that the manipulation can be used to represent the secret data. There are a number of methods that can be used implement text steganography such as: Line Shift Technique, Feature Technique, Word Shift Technique, and White Space Manipulation Technique.

- Image Steganography: For image steganography, an image is used as a carrier in which data is embedded. The idea used here is to change the pixel intensities by a minor factor such that it can carry secret data without being significantly distorted. The matrix used to define an image contains large number of bits which can be altered to achieve image based steganography.
- Audio Steganography: This type of steganography uses audio files such as .mp3, .wav formats as carriers to embed data into. The different techniques used to implement this type are: Low Bit Encoding, Echo Hiding, and Spread Spectrum.
- Video Steganography: Video is a collection of images and a typical video file will have .mov, .mp4, .avi, etc. extension. In this type of steganography the secret data is hidden in the images which collectively make a video. The most common method used to implement it is altering the values using discrete cosine transform (DCT).
- Network or Protocol Steganography: The available networking protocols such as TCP, UDP, ICMP and IP are used as carriers to implement protocol steganography. The OSI networking model has channels where steganography can be used.

## 4. IMAGE STEGANOGRAPHY

A computer interprets an image as a matrix of numbers that represent different intensity of pixels in different areas of an image. This digital form of an image where large number of bits is used to define it allows an image to be used as a carrier to achieve steganography. Since the bits represent intensity of pixels, they can be altered by a not so significant amount such that secret data is hidden into it and at the same time it cannot be noticed by any unwanted node in the middle of its path from source to destination. The general process for implementing image steganography involves embedding the secret data into a transport medium, called the carrier, which in this case is an image. This secret data after being embedded in the carrier image forms a steganography object which also may include a key that defines the encryption and decryption technique. The steganography key is used to encrypt the secret data before embedding it into the carrier and is also used to decrypt the secret data when it reaches its destination. The encryption of data provides more security in the communication process. The following diagram explains the general process for implementing image steganography:

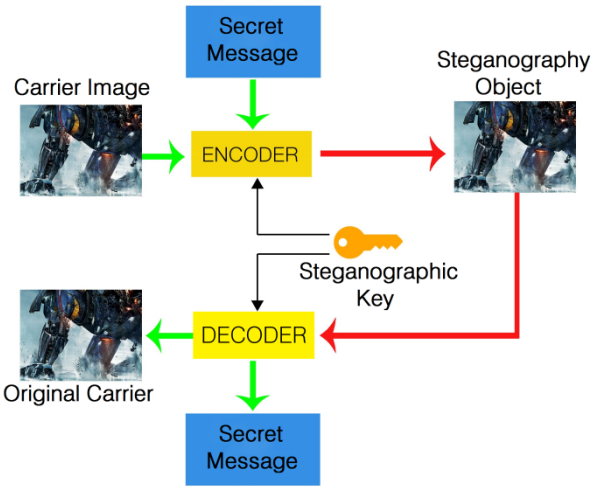


**Fig 1: Diagram for the process of Image Steganography**

The diagram shows that the secret message is encrypted using the steganography key and then embedded in the carrier image by an encoder to form a steganography object which is transmitted over a channel to the desired destination where a decoder uses the steganography key to decrypt the secret message from the carrier giving the secret message and the original carrier separately as its output.

Steganography Object = Secret Message + Carrier Image
+ Steganography Key

## 5. LEAST SIGNIFICANT BIT METHOD

One of the simplest techniques used to embed secret information in the carrier image is the Least Significant Bit (LSB). The technique used here is to directly put the secret information into the least significant bit plane of the host image. This change in the least significant bit has an impact on the cover image by a very slight amount which is not noticeable to the human eye. However, an important thing to keep in mind here is that a lossless compression algorithm is used for the steganography object so that no data is lost. If a lossy compression algorithm is used to compress the steganography object, the data embedded in it will get lost and the created steganography object will not serve the purpose of steganography. When a cover image of 24-bit colour is used wherein each byte is reserved for red, blue and green components of the image, the least significant bit of each byte can be used to store the secret data which means that a total of 3 bits can be stored in each pixel.

For example, the following shows the binary representation of 3 pixels of a 24-bit image where each set of 8-bits is reserved for the red, blue and green components of the image:

red blue green

00100001 00010111 01110101 – pixel 1

01010110 01110100 11010110 – pixel 2

00101000 00001001 01011111 – pixel 3

The least significant bit of each component is underlined in the above example which is to show that these bits can be used to store the secret data. Suppose a secret message "MISSISSIPPI RIVER" is to be embedded in a cover image. This message has 17 characters in it and the binary representation of this secret message is -

01001101 01001001 01010011 01010011 01001001
01010011 01010011 01001001 01010000 01010000
01001001 00100000 01010010 01001001 01010110
01000101 01010010 = 17 x 8 bits = 136 bits

Each pixel in a 24-bit color image can store 3 bits of data so for storing 136 bits of secret message 46 pixels of the cover image are needed to be altered. Thus, the least significant bits of 46 pixels of the cover image can be used to store the secret information and in this way image steganography can be implemented using the Least Significant Bit Method (LSB).

## 6. IMPLEMENTATION USING HUFFMAN CODING ALGORITHM

Huffman coding is a lossless data compression algorithm. This means no loss of data occurs when this compression algorithm is used. When Huffman coding is applied to a set of inputs, codes of variable length are generated and assigned to each character of the input set depending upon their frequencies. The frequency dependence of the assigned code on the character means that frequent characters in the given input have smaller codes assigned to them and the least frequent characters have larger codes assigned to them. In terms of bits, it means that lower number of bits is used to encode data that occurs more frequently and more number of bits for data that occurs less frequently. These variable length codes are assigned in such a way that the code of one character is not the prefix of any other character thereby ensuring that no ambiguity occurs while decoding the data. The assigned codes are kept in a code book which is made for each image or a set of images. In order to decode the data the code book needs to be supplied with the encoded data.

Since there is no fixed bits usage by each character, the resulting number of bits that is needed to be embedded in the cover image is significantly less. The same example "MISSISSIPPI RIVER" as the secret message can be used to see the compression rate achieved when Huffman coding is applied. Following steps are involved in applying Huffman coding to the secret message:

Building a Huffman tree -

- Create a leaf node for each distinct character in the message along with their frequencies.
- Sort the nodes in ascending order.
- Select two nodes with the minimum frequency and make a parent node of them.
- Assign the sum of the frequencies of the child nodes to the parent node.
- Repeat steps 2, 3 and 4 until there are no nodes left.
- When the complete tree is formed, assign code 0 to all the left and code 1 to all the right branches.

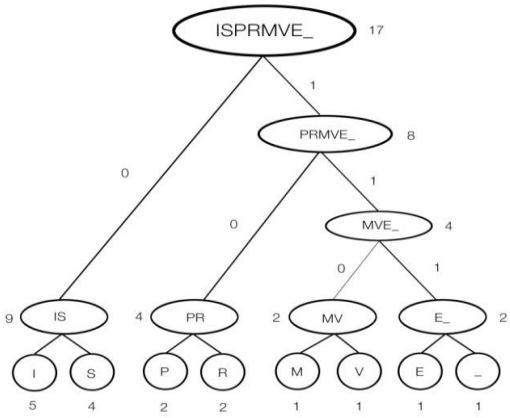


**Fig 2: Huffman Tree for message "MISSISSIPPI RIVER"**

The tree shown above is the Huffman tree for the secret message "MISSISSIPPI RIVER". The frequency of each node is mentioned next to it and all the left braches have code 0 while all the right branches have code 1 written next to them. This tree is used to assign codes to all the distinct characters by traversing the path from root to every leaf node. Following table shows the code assigned to each distinct character in the secret text and total bits used by each character:

**Table 1. Huffman Codes assigned to each character**

| **Character** | **Code Assigned** | **Frequency** | **Total Bits Used** |
|---|---|---|---|
| M | 1100 | 1 | 4 x 1 = 4 |
| I | 00 | 5 | 2 x 5 = 10 |
| S | 01 | 4 | 2 x 4 = 8 |
| P | 100 | 2 | 3 x 2 = 6 |
| R | 101 | 2 | 3 x 2 = 6 |
| V | 1101 | 1 | 4 x 1 = 4 |
| E | 1110 | 1 | 4 x 1 = 4 |
| _(space) | 1111 | 1 | 4 x 1 = 4 |
| **TOTAL** | | **17** | **46** |

The given table implies that the total number of bits needed to embed the message "MISSISSIPPI RIVER" in a cover image is 46. Since each pixel of a 24-bit colour image can store 3 bits of information, the total number of pixels needed to store 46 bits of secret information will be 16. It can be inferred that when secret message is embedded using LSB approach with the binary values of each character, a total of 136 bits are used instead when Huffman coding is applied to the secret message to assign variable length codes, the usage cuts down to a total of 46 bits.

The following table sums up the above proven points and compares both approaches on the basis of the example used:

**Table 2. Comparison between LSB and Huffman Coding technique**

| **Approach Used** | **Total bits needed to embed the secret message** | **Number of pixels needed in a 24-bit colour carrier image** |
|---|---|---|
| Least Significant Bit (LSB) with | | |

| | | |
|---|---|---|
| binary values of each character in text | 136 | 46 |
| Least Significant Bit (LSB) with Huffman Coding | 46 | 16 |

Hence, the compression rate achieved using the Huffman coding is -

**Total number of bits saved = 136 – 46 = 90 bits**

**% Compression = (90 / 136) x 100 = 66.17 %**

# 7. CONCLUSION

In this paper, steganography and its various types were briefly introduced. Image steganography is explained in detail and the generic process of encoding and decoding the secret data into and from the cover image is seen. 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. The use of Least Significant Bit with Huffman Coding is shown and its comparison with the former mentioned method is done. A conclusion can be made from the results shown above that when 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. Thus, it can be concluded that by using the proposed approach more efficient steganography can be implemented in terms of compression ratio.

# 8. REFERENCES

[1] Joan Condell, Kevin Curren, Paul Mc Kevitt "Digital image steganography: Survey and analysis of current methods", ELSEVIER, Signal Processing, Volume 90, Issue 3, March 2000, pages 727-752.

[2] N. F. Johnson and S. Jajodia, "Exploring steganography: seeing the unseen", IEEE Computer, 31(2)(1998), pp. 26-34.

[3] "Greedy Algorithms" - http://www.geeksforgeeks.org/greedy-algorithms-set-3-huffman-coding.

[4] Bin Li, Junhui He, Jiwu Huang, Yun Qing Shi, "A Survey On Image Steganography and Steganalysis", Journal of Information Hiding and Multimedia Signal Processing, Volume 2, Number 2, April 2011, ISSN 2073-4212.

[5] Rajarathnam Chandramouli, Mehdi Kharrazi, Nasir Memon, "Image Steganography and Steganalysis: Concepts and Practice", Springer Berlin Heidelberg, Digital Watermarking, Volume 2939, Computer Science, pp 35-49.

[6] T. Morkel, J.H.P. Eloff, M.S. Olivier, "An Overview Of Image Steganography", Information and Computer Security Architecture (ICSA) Research Group "http://mo.co.za/open/stegoverview.pdf".

[7] Jasleen Kour, Deepankar Verma, "Steganography Techniques – A Review Paper", International Journal of Emerging Research in Management and Technology, ISSN: 2278-9359, Volume 3, Issue 5.

[8] Shashikala Channalli, Ajay Jadhav, "Steganography An Art of Hiding Data", International Journal on Computer Science and Engineering, Volume 1(3), 2009, pp. 137-141.

[9] Arvind Kumar, Km. Pooja, "Steganography – A Data Hiding Technique", International Journal of Computer Applications, Volume 9 – No. 7, 2010.

[10] Shawn D. Dickman, "An Overview of Steganography", James Madison University Infosec Techreport, "http://vanilla47.com/PDFs/Cryptography/Steganography/An%20Overview%20of%20Steganography.pdf".

[11] Muhalim Mohamed Amin, Subariah Ibrahim, Mazleena Salleh, Mohd. Rozi Katmin, "Information Hiding Using Steganography", Universiti Teknologi Malaysia, "http://avierfjard.com/PDFs/Cryptography/Steganography/INFORMATION%20HIDING%20USING%20STEGANOGRAPHY.pdf".

[12] Nagham Hamid, Abid Yahya, R. Badlishah, Osamah M. Al-Qershi, "Image Steganography Techniques: An Overview", IJCSS, volume6, Issue3, IJCSS-670.

[13] Chi-Kwong Chan, L.M. Cheng, "Hiding data in images by using simple LSB", Elsevier, Pattern Recognition, Volume 37, Issue 3, March 2004, pages 469-474.

[14] R. Amirtharajan, R. Akila, P. Deepikachowdavarapu, "A Comparative Analysis of Image Steganography", "http://citeseerx.ist.psu.edu/viewdoc/download?doi=10.1.1.206.4715&rep=rep1&type=pdf".

[15] Chin-Chen Chang, Min-Hui Lin, Yu-Chen Hu, "A Fast and Secure Image Hiding Scheme Based on LSB Substitution", International Journal of Pattern Recognition and Artificial Intelligence, Volume 16, Issue 04, June 2002, DOI: 10.1142/S0218001402001770.