International Journal of Science and Research (IJSR)
ISSN: 2319-7064
ResearchGate Impact Factor (2018): 0.28 | SJIF (2018): 7.426

Image Steganography with Text Compression

Akanshi Srivastava[1]*, Vansh Badkul [2], Lohan Koka[3]

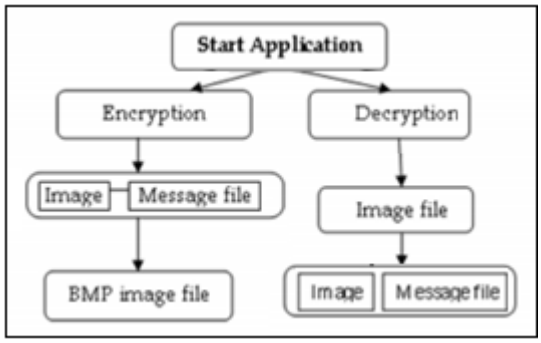
[1, 2, 3] School of Computer Sciences, VIT University, Vellore-632014, India

Abstract: ***Steganography is the practice of concealing messages or information within other non-secret text or data. Steganography maintains the secrecy between the two communicating parties. Image steganography is the practice of hiding data behind the cover image. The data hiding behind the image is hidden character-wise behind the pixels of the image. Therefore, whereas cryptography protects the contents of a message, steganography protects both the messages and communicating parties. In this paper, we review how we can compress and encrypt the data behind an image using various techniques and send it to the other person.***

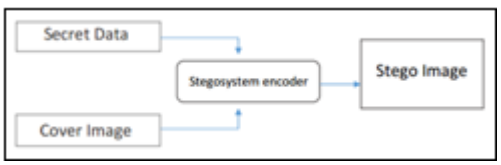
Keywords: Text Compression, Huffman Method of compression, Text Encryption, Image Steganography

1. Introduction

In today's world, communication has become a basic necessity in our life. Communication is nothing but the transfer of data between two different parties. The protection of this data is a primary concern for both the sender and the receiver. The word steganography is derived from the Greek words stego meaning "cover" and grafia meaning "writing" defining it as "Covered writing". The main benefit of the steganography over cryptography is that the intended secret message, data or piece of information does not draw attention to itself as an object of scrutiny. One of the reasons why the usage of steganography does not prevail over cryptography is due to the limitations in the size of the data. In this paper, we try to solve this problem by compressing the size of the data(text) using techniques like Huffman coding. Huffman coding is a lossless data compression algorithm. The proposed idea is to assign variable-length codes to input characters, lengths of assigned codes are based on frequencies of corresponding characters. The least frequent character gets the largest code and the most frequent character gets the smallest code.



Terminology
Secret data: Message that needs to be hidden within an image.
Cover image: It is the carrier that hides the message in the image.
Stego image: The image carrying the secret data.



Cryptography versus Steganography

Cryptography:
- It is the study of hiding information.
- The existence of the encrypted message is visible to the world.
- It only protects the content of the message.
- It attracts unwanted attention.

Steganography [1]:
- It is the practice of concealing information within other data.
- Only the sender and the receiver are aware of the presence of the encrypted message.
- It protects both the content and the message.
- It doesn't attract any attention.

Image Steganography technique
Least Significant Bit (LSB): This is a technique used for image steganography. It is mostly used for hiding data. In this method, the data is embedded within the image by replacing the least significant bits of the image pixel with bits of secret data. The computational complexity of this method is very low.

Image Compression
Image compression is the minimization of the size of the image without completely compromising the quality of the image. This reduction in the size of the image allows us to store more images in our memory disk. Lossless Compression: In this type of compression there is no loss of data from the original file. Lossy Compression: In this type of compression there is some amount of lost data from the original file.

Huffman Coding [6]
Huffman coding is based on the frequency of occurrence of a data item. The principle is to use a lower number of bits to encode the data that occurs recurrently. Codes are stored in a 'Code Book' which may be constructed for each image or a set of images. This codebook plus encoded data must be transmitted to enable decoding.

The variable-length codes assigned to the input characters are Prefix Codes, means the codes (sequences of bits) are then assigned in such a way that the code assigned to one

Licensed Under Creative Commons Attribution CC BY

character is not the prefix of code assigned to any other character. Therefore, Huffman Coding makes sure that there is no ambiguity when decoding the generated bitstream.

2. Proposed Method

In this paper, we will make software through which we can compress the messages efficiently and encrypt messages or hide the messages for security.

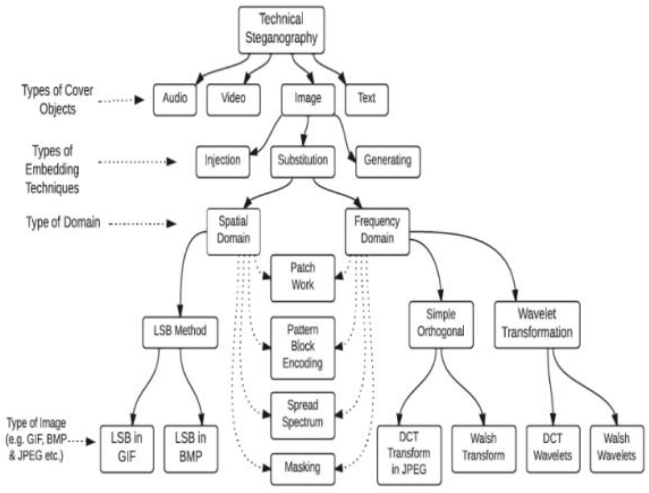
The main advantage of steganography is that the intended discrete message or information does not attract attention to itself as an object of scrutiny. We'll use Huffman coding for text compression.

The information which is to be transmitted over the image is converted to 0's and 1's form by Huffman coding. This converted code is embedded inside the image by varying the Least Significant Bit (LSB)[9][10] of each of the pixel values of information can be decrypted by Huffman Table which is embedded in envelop image.

The information can be decrypted by Huffman Table which is embedded in image itself so that the image becomes impartial information to the viewer.

The following steps will explain further-

1) In this project, we first encrypt the plaintext to generate the ciphertext, and then embed the ciphertext in an image.
2) The generated stegoimage is sent over to the intended recipient.
3) If a third party snoops the stegoimage in between, then they will just see some harmless-looking picture.
4) Once the recipient receives the stegoimage, the ciphertext is extracted from it by reversing the logic that was used to embed it in the first place.
5) The ciphertext is decrypted using the traditional cryptography to get back the original plaintext.



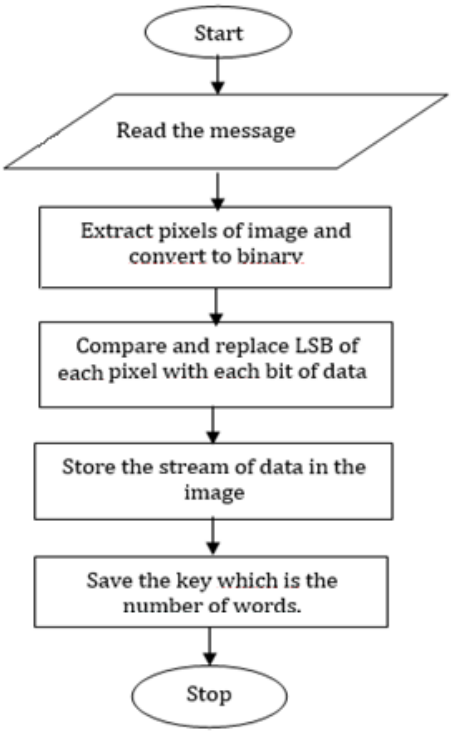
Component Design

At first, we use the compression and encryption technique to change the plain text which can be concealed in the image. Then image libraries are used to manipulate the image which is required to be stegnographed[8]. Then we use the ASCII and color functions to extract the LSB[5] of each pixel which can be compared with the data and modified to store the data in it. Each data is decrypted to the specified technique and then converted into binary. This can be decrypted on the receiver side with the given keyword which is generated during the time of steganography. This keyword takes different logic to get generated. In receiver side modification in the image can't be seen through the naked eye, which helps us to save our data from damage.

Algorithms of the proposed work

Encryption:

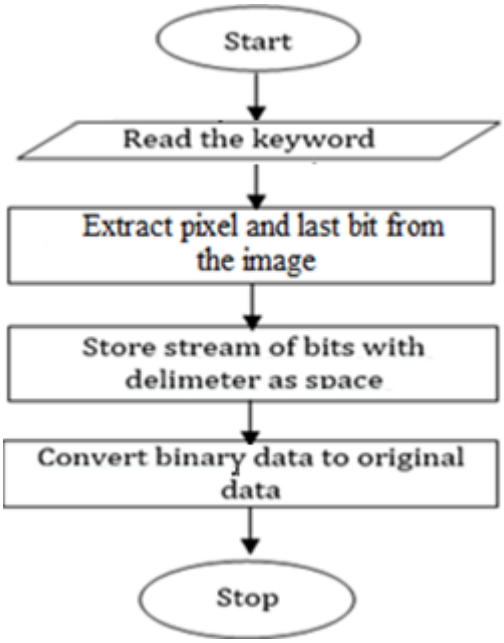
1) Check whether the picture is in the proper format (PNG) according to the program.[4]
2) Get the data from the user and convert it into binary stream of data.
3) Use for loops to extract each pixel value and convert it into binary.
4) Now compare each bit of data and the last bit of 'B' value in binary must be taken.
5) Replace the last bit of each pixel with each bit of data.
6) Run the for loop until the complete stream of data is stored inside the image
7) Save the image in a separate name. and send the number of words as the key.



Decryption:

1) Run for loops with a range of the sent keyword.
2) Start extracting each pixel and extract the last bit of the tuple.
3) Separate the stored stream of bits with the delimiter as space.
4) Use the functions of python to convert the binary data into original data.
5) The key plays a vital role in decrypting the data. The larger the key the harder it becomes for the intruder to decrypt the data.

Licensed Under Creative Commons Attribution CC BY



Pseudo Code

Data Embedding [7]
The embedding process is as follows.
Inputs Cover image, stego-key and the text file Output stego image [2]

Procedure
Step 1:Extract all the pixels of the cover image.
Step 2: Extract the characters from the Stego key.
Step 3: Choose the first pixel and pick characters of the Stego key
Step 4: Place it in the first component of the pixel.
Step 5: Place some terminating symbol to indicate the end of the key. 0 has been used as a terminating symbol in this algorithm.
Step 6: Insert characters of the text file in each the first component of next pixels by replacing it.
Step 7: Repeat step 6 till all the characters have been embedded.
Step 8: Again place some terminating symbol to indicate the end of data.
Step 9: Obtained stego image.

Data Extraction
The extraction process is as follows.
Inputs: Stego-image file, stego-key
Output: Secret text message.

Procedure:
Step 1: Extract the pixels of the stego image.
Step 2: Now, start from the first pixel and extract stego key characters from the first component of the pixels. Follow Step3 up to terminating
symbol, otherwise follow step 4.
Step 3: If this extracted key matches with the key entered by the receiver, then follow Step 5, otherwise terminate the program.
Step 4: If the key is correct, then go to next pixels and extract secret message characters from the first component of the next pixels. Follow
Step 5 till up to terminating symbol, otherwise follow step 6.
Step 6: Extract the secret message.

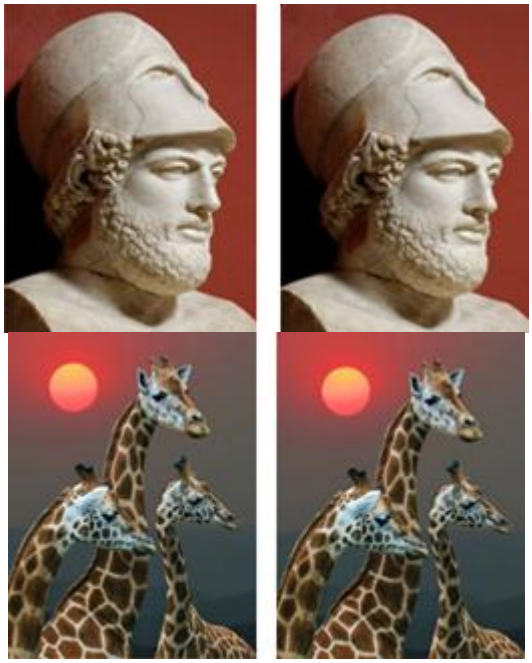
Image Encoding Algorithm
An inputs Image file, stego key, and image file Output Stego image.[3]
1) The cover and secret images are read and converted into the unit8 type.
2) The numbers in secret image matrix are conveyed to 8-bit binary. Then the matrix is reshaped to a new matrix.
3) The matrix of the cover image used is reshaped as well.
4) Perform the LSB technique described on this matrix.
5) The stego-image, which is very similar to the original cover image, is achieved by the reshaping matrix.
6) While extracting the data, the LSB of the stego image is collected and they are reconstructed into the decimal numbers. The decimal numbers are reshaped to the secret image.

Comparison with existing studies and methods:
- This technique uses RGB[11] true color images for embedding process[12] .
- Data is encrypted then it is embedded into the image using steganographic methods.
- A new improved version of LSB image steganography based on efficient filtering technique using status bits.
- This uses pixel adjustment technique for better stego image quality.
- This results in high hidden capacity.
- The Huffman tree can also be used for image steganography.
- It is hard for the attacker to extract the data as the Huffman table reduces the cover image size.

Description of the results obtained through proposed approach
The image will also ask for the key for decryption, which can be sent separately. If proper key is not provided the data inside the image will automatically be damaged. The hiding of data was properly executed. It is user wish to generate a key for the protection.

Results obtained through the proposed method:

Licensed Under Creative Commons Attribution CC BY



1. Original Images **2. Stego Images**

3. Conclusion

In this paper we have proposed a new method of image steganography: The quality of the image or the steganography object should not be change upon adding excess data. The data should be undetectable without the key generated. The secrete data should survive attacks by the intruders. In this only alphabets can be used. This helps us to keep the data secured more because when the image is stolen, the code for decryption will yield a result with special characters and numbers. This will not be the actual information. The actual information can be extracted only with proper conditions on the decrypting code.

References

[1] Kesslet, Gary C. *An Overview of Steganography for the Computer Forensics Examiner*, Burlington, 2004.
[2] Fridrich, J., R. Du, M. Long: Steganalysis Of LSB Encoding In Color Images, Binghamton, 2007.
[3] Vehse, Heymo. YAVI: *Yet Another Vigenere Algorithm* www.leafraker.com/yavi
[4] Mohammad Shirali-Shahreza , "*A new method for real time steganography",* ICSP 2006 Proceedings of IEEE .
[5] Ravi shah , Abhinav Agraval & subramaniam Ganesham*, "Frequency domain real time digital image watermarking "* Oakland university.
[6] A. Nag, J.P. Singh, S. Biswas, D. Sarkar, and P.P. Sarkar*, "A Huffman Code Based Image Steganography Technique",* 1st International Conference on Applied Algorithm (ICAA) Jan. *2014, pp. 257-265.*
[7] H. Yang, X. Sun and G. Sun*, "A High-Capacity Image Data Hiding Scheme Using Adaptive LSB Substitution",* Journal of Radio Engineering, Vol. 18, No. 4, pp. 509-516, 2009
[8] *D. Samidha and D. Agrawal, "Random Image Steganography in Spatial Domain"* IEEE International Conference on Emerging Trends in VLSI, Embedded System, Nano Electronics and Telecommunication System (ICEVENT), Jan. 2013, pp. 1-3.
[9] Jain, Nitin, Sachin Mesh ram, and Shikha Dubey. *"Image Steganography Using LSB and Edge–Detection Technique.* "Internationala Journal of Softcomputing and Engineering (IJSCE) ISSN (2012): 2231-2307
[10] *"Image Steganography Least Significant Bit with Multiple Progressions"*Savita Goel.
[11] D. Debnath, S. Deb, N. Kar, *"An Advanced Image Encryption Standard Providing Dual Security: Encryption Using Hill Cipher and RGB Image Steganography"*, IEEE International Conference on Computational Intelligence and Networks (CINE), Jan. 2015, pp. 178-183.
[12] D. Debnath, S. Deb, N. Kar, *"An Advanced Image Encryption Standard Providing Dual Security: Encryption Using Hill Cipher and RGB Image Steganography"*, IEEE International Conference on Computational Intelligence and Networks (CINE), Jan. 2015, pp. 178-183

Licensed Under Creative Commons Attribution CC BY