LSB Steganography with PSNR and Data Integrity Check

Shivam Kesarwani[1], Nitin Pal[2], Mayank Negi[3], Devansh Singh[4], Aseem Aggarwal[5], Tarun[6]

[1,2,3,4,5,6]*Dept. of Information Technology, GBPUA&T Pantnagar, Uttarakhand, India*

Abstract - *With the increasing widespread use of information technologies, work and transactions are shifted to electronic environments and it becomes important to protect or secure the information stored, processed and transferred in these environments. In an environment where digital data communication occurs, there are many threats for the message sent such as unauthorized access, damage, destruction, modification and reproduction. Despite the precautions taken, reports for these threats are increasing day by day. Various techniques have been developed to remove these threats in response to the emergence of these threats. The techniques likes cryptography, watermarking, steganography have been able to provide some relief. Steganography of these entire have recently caught the highlight and constant research and development is taking place with steganography as centre point.*

This paper aims at an effective data hiding technique i.e. steganography based on LSB insertion and to calculate the peak signal to noise ratio (PSNR) between the encrypted image and original image. It also uses SHA to check the data integrity of the extracted data from the image.

Key Words: **LSB, Steganography, PSNR, SHA, Data Integrity, MSE**

1. INTRODUCTION

Steganography is one of the most powerful techniques to conceal the existence of hidden secret data inside a cover object. Images are the most popular cover objects for Steganography and in this work image steganography is adopted. Steganography includes the concealment of information within computer files. In digital steganography, electronic communications may include steganographic coding inside of a transport layer, such as a document file, image file, program or protocol. Media files are ideal for steganographic transmission because of their large size. Steganography is the study of embedding and hiding messages in a medium called a *cover text*.

1.1 Existing System

Privacy is one of the most important issues in today's communication systems. In applications where the importance of your privacy is indispensable, the main aim is to send the information to desired target without being captured by the third persons or by bringing them in such a way that they cannot understand. Today, researchers have developed data hiding methods using a wide variety of digital media. At this point, it is important not only to hide data, but also to develop mechanisms to prevent third parties from identifying hidden data.

1.2 Proposed System

The proposed LSB Steganography project tells us about that how a data can be hidden in an image. It uses LSB to implement Steganography. LSB algorithms have a choice about how they embed that data to hide. They can embed losslessly, preserving all information about the data, or the data may be generalized so that it takes up less space. In this embedding process, 4 out of 8 bits have changed. However, when the embedding is done in order, an image in which the message is hidden can be easily solved by third parties. Since the same random number is likely to be generated more than once, there is a possibility that the same group of pixels may be changed more than once in the random embedding process. In this case, character loss can be found when the hidden message is solved.

Image quality is a characteristic for an image that measures the apparent image debasement (regularly, contrasted with a perfect or ideal image). Imaging systems may present some amounts of artifacts or distortion in the image, so the quality assessment is an essential issue. One of the most well-known and widely used measure of comparing two images is the Peak Signal to Noise Ratio (PSNR). PSNR is an engineering term for the proportion of the maximum possible power of original image to the power of the differences between original image and stego image.

2. IMPLEMENTATION

2.1 LSB algorithm

LSB (Least Significant Bit) substitution is the process of adjusting the least significant bit pixels of the carrier image. It is a simple approach for embedding message into the image. The Least Significant Bit insertion varies according to number of bits in an image. For an 8 bit image, the least significant bit i.e., the 8th bit of each byte of the image is changed to the bit of secret message. For 24 bit image, the colors of each component like RGB (red, green and blue) are changed. LSB is effective in using BMP images since the compression in BMP is lossless. But for hiding the secret message inside an image of BMP file using LSB algorithm it requires a large image which is used as a cover. LSB substitution is also possible for GIF formats, but the problem with the GIF image is whenever the least significant bit is changed the whole color palette will be changed. The problem can be avoided by only using the gray scale GIF images since the gray scale image contains 256 shades and the changes will be done gradually so that it will be very hard to detect. There are many approaches available for hiding the data within an image: one of the simple least significant bit submission approaches is "Optimum Pixel Adjustment Procedure‟. The simple algorithm for OPA explains the procedure of hiding the sample text in an image.

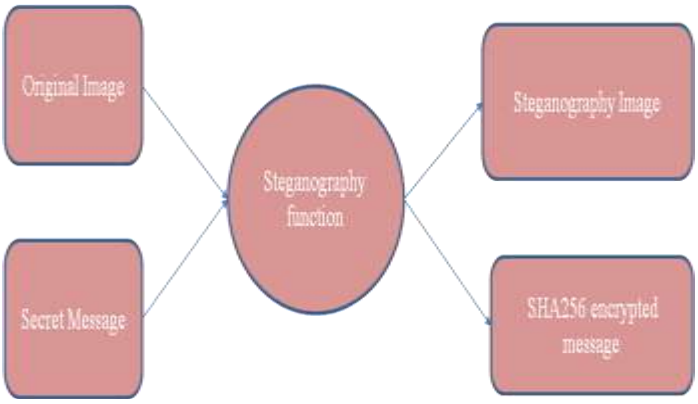

Fig -1: Encryption

Step1: A few least significant bits (LSB) are substituted with in data to be hidden.

Step2: The pixels are arranged in a manner of placing the hidden bits before the pixel of each cover image to minimize the errors.

Step3: Let n LSBs be substituted in each pixel.

Step4: Let d= decimal value of the pixel after the substitution.

d1 = decimal value of last n bits of the pixel.

d2 = decimal value of n bits hidden in that pixel.

Step5: If (d1~d2)<=(2^n)/2

Then no adjustment is made in that pixel.

Else

Step6: If(d1<d2)

d = d –2^n.

If(d1>d2)

d=d+2^n.

This "d‟ is converted to binary and written back to pixel.

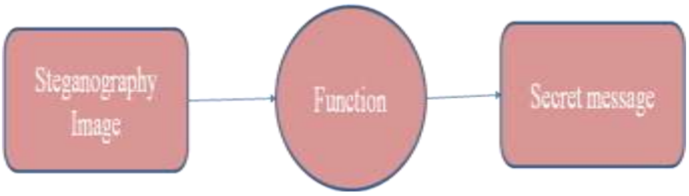

Fig -2: Decryption

2.2 PSNR

To compute the PSNR, the block first calculates the mean-squared error using the following equation: In the previous equation, M and N are the number of rows and columns in the input images, respectively. Then the block computes the PSNR using the following equation:

$$PSNR = 10\ Log_{10}\ (R^2/MSE)$$

In the previous equation, R is the maximum fluctuation in the input image data type. For example, if the input image has a double-precision floating point data type, then R is 1. If it has an 8-bit unsigned integer data type, R is 255, etc.

The Mean Square Error (MSE) represents the cumulative squared error between the compressed and the original image, whereas PSNR represents a measure of the peak error.

The lower the value of MSE, the lower the error.

$$MSE = \frac{\Sigma_{M,N}[I_1(M.N) - I_2(M.N)]^2}{M * N}$$

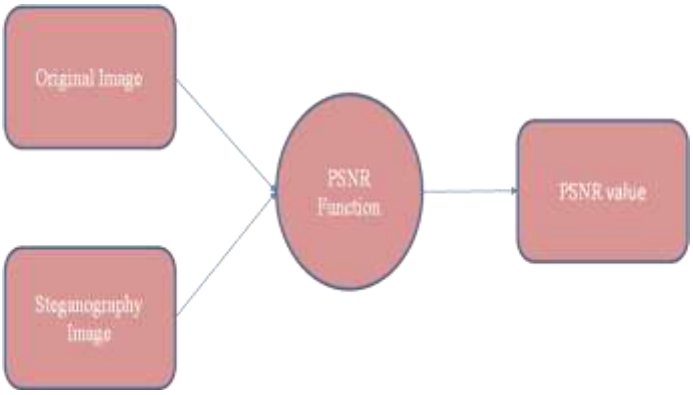

Fig -3: PSNR Calculation

Sometimes PSNRs vary wildly between two almost indistinguishable images; similarly you can have two images with the same PSNR where there is a very obvious difference in quality. The structural similarity index measurement (SSIM) and some of its variations are generally considered better from this perspective, but still not perfect models for human perception.

High PSNR means good image quality and less ERROR introduced to the image.

2.3 SHA

SHA, (Secure Hash Algorithms) are set of cryptographic hash functions defined by the language to be used for various applications such as password security etc. Some variants of it are supported by Python in the "**hashlib**" library. These can be found using "algorithms_guaranteed" function of hashlib.

Functions associated:

- encode() : Converts the string into bytes to be acceptable by hash function.
- hexdigest() : Returns the encoded data in hexadecimal format.

SHA is a cryptographic hash function. A hash function takes an initial unencrypted text, called the plaintext, and produces a theoretically unique number that constitutes the encrypted message. SHA creates a 160-bit number, which is a number between 0 and 1.46 x 10^48. It is not possible for this number to be guaranteed unique for all possible plaintext messages, as the number of such messages is theoretically infinite, but the odds are approximately 2^80, or 1.21 x 10^24, against two messages producing the same encrypted result. If this does occur, this is called a collision. A collision provides a mathematical attack on an encryption algorithm, making it possible for a cryptographer to decrypt the plaintext.

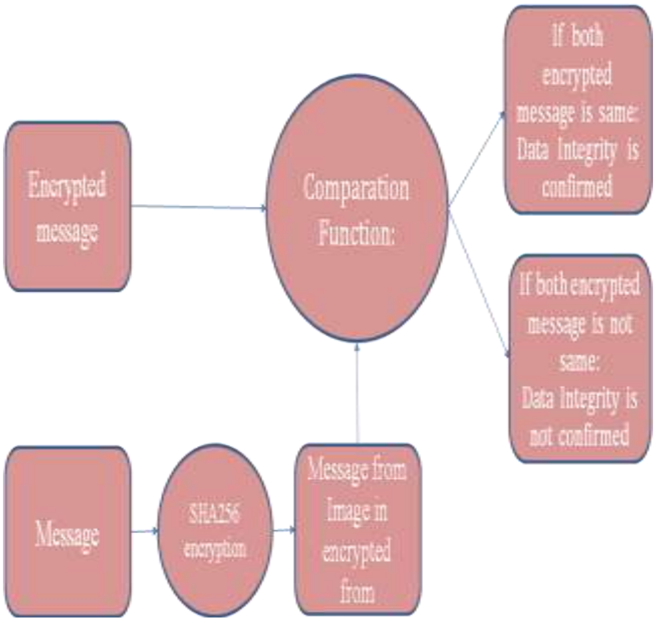
Encrypted message
Comparation Function:
If both encrypted message is same: Data Integrity is confirmed
If both encrypted message is not same: Data Integrity is not confirmed
Message
SHA256 encryption
Message from Image in encrypted from

Fig -4: Data Integrity Check

3. CONCLUSION

Hiding a message with steganography method reduces the chance of message being detected. In and of itself, steganography is not a good solution to secrecy, but neither is simple substitution and short block permutation for encryption. But if these methods are combined, you have much stronger encryption routines. Like any tool, steganography is neither inherently good nor evil, it is the manner in which it is used which will determine whether it is a benefit or a detriment to our society.

REFERENCES

[1] Ming, Chen, Z. Ru, N. Xinxin, and Y. Yixian, "Analysis of Current Steganography Tools:Classifications & Features", Information Security Beijing University of Posts & Telecommunication, Beijing, December 2006. Centre.

[2] Chandramouli R and Memon N, "Analysis of LSB based image steganography techniques", Proceedings 2001 International Conference on Image, Vol. 3, pp. 1019-1022.

[3] Implementing Cisco IOS Network Security (IINS) Catherine Piquet Copyright © 2009 Cisco Systems, Inc. Published by: Cisco Press 800 East 96th Street Indianapolis, IN 46240 USA.

BIOGRAPHIES

Shivam Kesarwani is currently pursuing Btech from GBPUA&T Pantnagar . He aims at becoming a good developer

Nitin Pal is currently pursuing Btech from GBPUA&T Pantnagar . He aims at becoming a good developer

Mayank Negi is currently pursuing Btech from GBPUA&T Pantnagar . His aim is to become a game developer

Devansh Singh is currently pursuing Btech from GBPUA&T Pantnagar . He aims at pursuing MBA

Aseem Aggarwal is currently pursuing Btech from GBPUA&T Pantnagar . He aims at pursuing MBA

Tarun Kharakwal is currently pursuing Btech from GBPUA&T Pantnagar . He aims at becoming a good developer