ISSN(Online): 2320-9801
ISSN (Print) : 2320-9798

International Journal of Innovative Research in Computer and Communication Engineering

(An ISO 3297: 2007 Certified Organization)

Vol. 4, Issue 5, May 2016

Proposed Image Security Using Combination of Steganography and Hashing Algorithm

Peeyush, Saurabh Saraswat

M.Tech Student, Dept. of ECE., BITS Bhiwani, MDU Rohtak, India

Assistant Professor, Dept. of ECE., BITS Bhiwani, MDU Rohtak, India

ABSTRACT: In this paper we are going to secure the image using combination of steganography and secure hashing Algorithm for high resolution images. Although the information of the image can be secured using steganography but because all the information in steganography is only hide not encrypted so to make it secure from intruder encryption is done. In this paper secure hash algorithm-256 is applied after the bit swapping algorithm to secure the image and result shows a mass improvement the only staggered image. There are two different procedures, which are used here at the sender's end and receiver's end respectively. The procedures are used here as the key of Data Hiding and Extraction.

KEYWORDS: Data hiding, DCT, security, multimedia, steganography.

I. INTRODUCTION

In the era of information technology and communication, Internet and digital media are getting more and more popular and requirement of secure transmission of data increasing day by day. Various good techniques are proposed and already taken into practice.

Hiding the data in a medium so the its not accessible for hacker to retrieve the information is known as Steganography. Steganography term came from the two Greek words Steganos and graphia, where "steganos" means covered or secret and "graphic" means writing or drawing. So it is the art of writing a message and put it in a cover. The main purpose of steganography is to hide the fact of communication. In this the sender embedded its message into the text, image, video, or audio file so that hackers will not be aware of the message. [1][2]

Data Hiding is the process of secretly embedding information inside a data source without changing its perceptual quality. Data Hiding is the art and science of writing hidden messages in such a way that no one apart from the sender and intended recipient even realizes there is a hidden message. Generally, in Data Hiding, the actual information is not maintained in its original format and thereby it is converted into an alternative equivalent multimedia file like image, video or audio which in turn is being hidden within another object. This apparent message is sent through the network to the recipient, where the actual message is separated from it.

The requirements of any data hiding system can be categorized into security, capacity and robustness Cox et al. (1996). All these factors are inversely proportional to each other creating the so called data hiding dilemma. The focus of this paper aims at maximizing the first two factors of data hiding i.e. security and capacity coupled with alteration detection.

There are different kind of secure hash algorithms named as, SHA-1, SHA-224, SHA-256, SHA-384, SHA-512, SHA-512/224 and SHA-512/256. All of the secure hash algorithms are iterative, one-way hash functions that can process a message to produce a condensed representation called a *message digest*. These algorithms enable the determination of a message's integrity: any change to the message will, with a very high probability, results in a different message digest. This property is useful in the generation and verification of digital signatures and message authentication codes, and in the generation of random numbers or bits.

Each algorithm can be described in two stages: preprocessing and hash computation. Preprocessing involves padding a message, parsing the padded message into *m*-bit blocks, and setting initialization values to be used in the hash computation. The hash computation generates a *message schedule* from the padded message and uses that schedule, along with functions, constants, and word operations to iteratively generate a series of hash values. The final hash value generated by the hash computation is used to determine the message digest.

The proposed scheme is a data-hiding method that uses high resolution digital image as a cover signal. The proposed recipient need only process the required steps in order to reveal the message; otherwise the existence of the hidden information is virtually undetectable. The proposed scheme provides the ability to hide a significant quality of information making it different from typical data hiding mechanisms because here we consider application that require significantly larger payloads like video-in-image and picture-in-image.

The purpose of hiding such information depends on the application and the needs of the owner/user of the digital media. Data hiding requirements include the following:

a. Imperceptibility- The image with data and original data source should be perceptually identical.
b. Robustness- The embedded data should survive any processing operation the host signal goes through and preserve its fidelity.
c. Capacity-Maximize data embedding payload.
d. Security- Security is in the key.

Data Hiding is the different concept than cryptography, but uses some of its basic principles [3].

In this paper, we have considered some important features of data hiding. Our consideration is that of embedding information into image, which could survive attacks on the network.

II. PREVIOUS WORKS

As image file consist of several image sequences, so considering the data hiding technique of image will also apply for video data hiding we get:

A. Least-significant bit modifications

The most widely used technique to hide data, is the usage of the LSB. Although there are several disadvantages to this approach, the relative easiness to implement it, makes it a popular method. To hide a secret message inside a image, a proper cover image is needed. Because this method uses bits of each pixel in the image, it is necessary to use a lossless compression format, otherwise the hidden information will get lost in the transformations of a lossy compression algorithm.

When using a 24 bit color image, a bit of each of the red, green and blue color components can be used, so a total of 3 bits can be stored in each pixel. Thus, a 800 ×600 pixel image can contain a total amount of 1.440.000 bits (180.000 bytes) of secret data. For example, the following grid can be considered as 3 pixels of a 24 bit color image, using 9 bytes of memory:

(00100111 11101001 11001000)
(00100111 11001000 11101001)
(11001000 00100111 11101001)

When the character A, which binary value equals 10000001, is inserted, the following grid results:

(00100111 11101000 11001000)
(00100110 11001000 11101000)
(11001000 00100111 11101001)

In this case, only three bits needed to be changed to insert the character successfully. On average, only half of the bits in an image will need to be modified to hide a secret message using the maximal cover size. The resulting changes that are made to the least significant bits are too small to be recognized by the human eye, so the message is effectively hidden.

While using a 24 bit image gives a relatively large amount of space to hide messages, it is also possible to use a 8 bit image as a cover source. Because of the smaller space and different properties, 8 bit images require a more careful approach. Where 24 bit images use three bytes to represent a pixel, an 8 bit image uses only one. Changing the LSB of that byte will result in a visible change of color, as another color in the available palette will be displayed. Therefore, the cover image needs to be selected more carefully and preferably be in grayscale, as the human eye will not detect the difference between different gray values as easy as with different colors.

B. Masking and filtering

Masking and filtering techniques, usually restricted to 24 bits or grayscale images, take a different approach to hiding a message. These methods are effectively similar to paper watermarks, creating markings in an image. This can be achieved for example by modifying the luminance of parts of the image. While masking does change the visible Properties of an image, it can be done in such a way that the human eye will not notice the anomalies.

Since masking uses visible aspects of the image, it is more robust than LSB modification with respect to compression, cropping and different kinds of image processing. The information is not hidden at the "noise" level but is inside the visible part of the image, which makes it more suitable than LSB modifications in case a lossy compression algorithm like JPEG is being used [4].

C. Transformations

A more complex way of hiding a secret inside an image comes with the use and modifications of discrete cosine transformations. Discrete cosine transformations (DCT)), are used by the JPEG compression algorithm to transform successive 8 x 8 pixel blocks of the image, into 64 DCT coefficients each. Each DCT coefficient *F(u,v)* of an 8 x 8 block of image pixels *f(x, y)* is given by:

$$F(u,v) = \frac{1}{4}C(u)C(v)\left[\sum_{x=0}^{7}\sum_{y=0}^{7} f(x,y) * cos\frac{(2x+1)u\pi}{16} cos\frac{(2y+1)v\pi}{16}\right]$$

Where *C(x) =1/√2* when *x* equals 0 and *C(x) =1* otherwise. After calculating the coefficients, the following quantizing operation is performed:

$$F^{Q}(u,v) = \left[\frac{F(u,v)}{Q(u,v)}\right]$$

Where *Q(u,v)* is a 64-element quantization table. A simple pseudo-code algorithm to hide a message inside a JPEG image could look like this:

```
Input: Message, cover image
Output: steganographic image containing message
While (data left to embed)
 Do
{
Get next DCT coefficient from cover image
 If DCT not equal to 0 and DCT not equal to 1 then
           {
             Get next LSB from message
             Replace DCT LSB with message bit
             }
End if
             Insert DCT into steganographic image                }
End while
```

Although a modification of a single DCT will affect all 64 image pixels, the LSB of the quantized DCT coefficient can be used to hide information. Lossless compressed images will be suspect able to visual alterations when the LSB are modified. This is not the case with the above described method, as it takes place in the frequency domain inside the image, instead of the spatial domain and therefore there will be no visible changes to the cover image [5].

When information is hidden inside image the program or person hiding the information will usually use the DCT (Discrete Cosine Transform) method. DCT works by slightly changing the each of the images in the image, only so much though so it's isn't noticeable by the human eye. To be more precise about how DCT works, DCT alters values of certain parts of the images, it usually rounds them up.

For example if part of an image has a value of 6.667 it will round it up to 7. Data Hiding in Video is similar to that of Data Hiding in Images, apart from information is hidden in each frame of the image. When only a small amount of information is hidden inside of image it generally isn't noticeable at all, however the more information that is hidden the more noticeable it will become.

III. PROPOSED WORK

The main high resolution video file is nothing but a sequence of high resolution image called frames. Initially I will like to read the image and convert the images into its pixels intensity format . The algorithm will work in following steps:

a. Firstly we will try to convert the image in a matrix of intensity matrix and convert the pixel information into its binary value.
b. Then we will flips the bit information by a random fashion and store it.
c. The staggered image will be encrypted using Secure Hash Algorithm (SHA256) and resulted image go to the processing of compression using block process.
d. The information will be converted into blocks of size [8 8] and processed through DCT compression algorithm.
e. The information resulted through DCT will be compressed via masking and filtering algorithm. The compressed information will contain compression ratio of 6-10 as per the number of DCT coefficient passed.
f. The reverse process will be used at receiver end.

Apparently, if the image sequence is large enough, the frame period can be accordingly large. The encoder reads these parameters from a file. The same file is read by the software that extracts the message, so as both of the two codes to be synchronized. After streaming the video file into frames I will like to use the above explained algorithm.

Recently [7] has claimed that LSB replacement involving more than one least significant bit planes is less detectable than single bit plane LSB replacement. Hence the use bit permutation of multiple bit planes for embedding has been encouraged. But the direct use of 3 or more bit planes leads to addition of considerable amount of noise in the cover image. So as my work is in high resolution image so I am getting a RGB combination of each pixel and permuting the bit value and encrypt the data and compress also.

IV. RESULTS

In the present work, the above mentioned algorithm is used to secure the image via matlab programming model and results related to PSNR and images at different stages of steganography and cryptography have been presented. Results obtained at different stages with PSNR have been explained below:

A. Original image and image after Steganography

The original image is of tagged image file format and its intensity vector is captured in a variable I. The size of image shown in figure 4.1 is of 256*256 pixels. The steganographic image obtained by swapping bits is shown in figure 4.2. The PSNR of Steganographic image is 8.4755, MSE is 9.2369e+0, MAXERR is 225 and L2RAT is 1.1848.

Figure 4.1 Original image Cameraman.tif

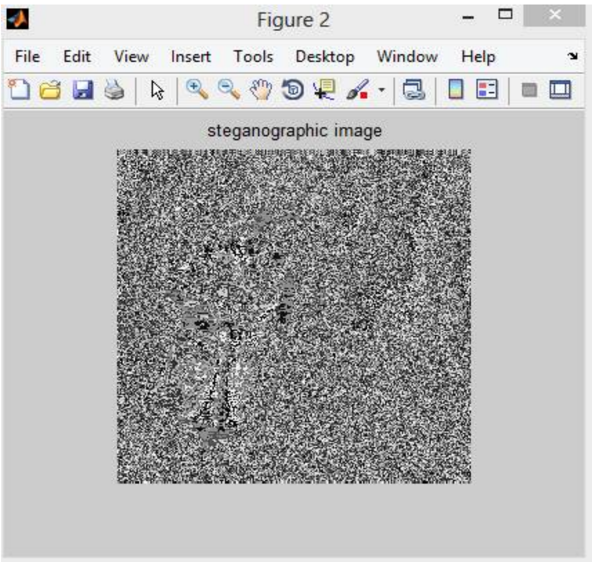

Figure 4.2 Steganographic image of cammerman.tif

B. Image after applying SHA on Steganographic image

Figure 4.3 shows that image after applying the secure hash algorithm based cryptography and results clearly evidence of improvement in security of image. The PSNR of the image is 5.1374 MSE is 9.4073e+03,MAXERR is 250.

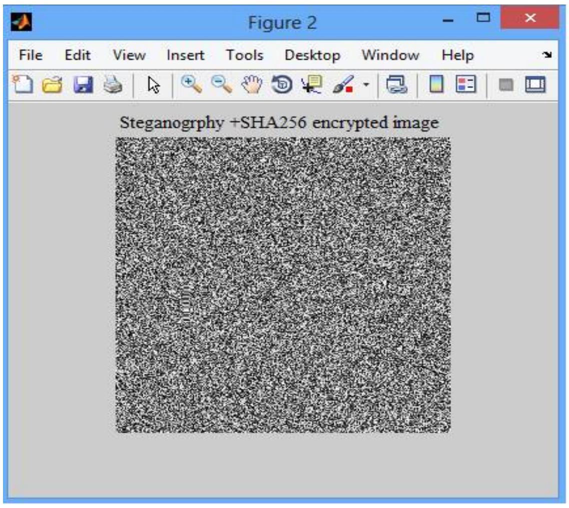
Figure 2
File Edit View Insert Tools Desktop Window Help
Steganogrphy +SHA256 encrypted image

Figure 4.3 steganographic +SHA256 encrypted image

C. STEGANOGRAPHIC IMAGE AFTER COMPRESSION

Figure 4.4 is the image after DCT Compression with compression ratio=10 and BPP equals to 6.2. The PSNR AFTER Compression with figure 4.2 and 4.1 is 4.8492 and 5.5861 respectively.

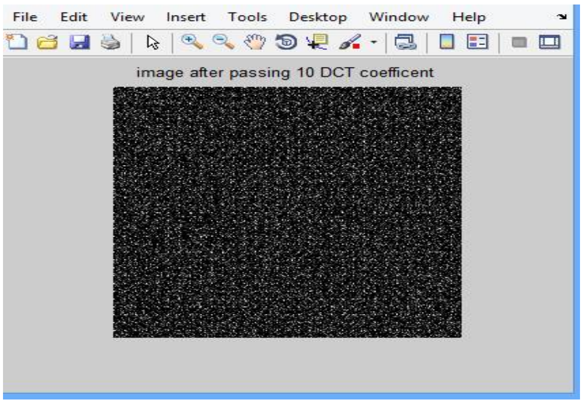
File Edit View Insert Tools Desktop Window Help
image after passing 10 DCT coefficent

Figure 4.4 Image after DCT Compression technique

D. Retrieval of Steganographic image using IDCT

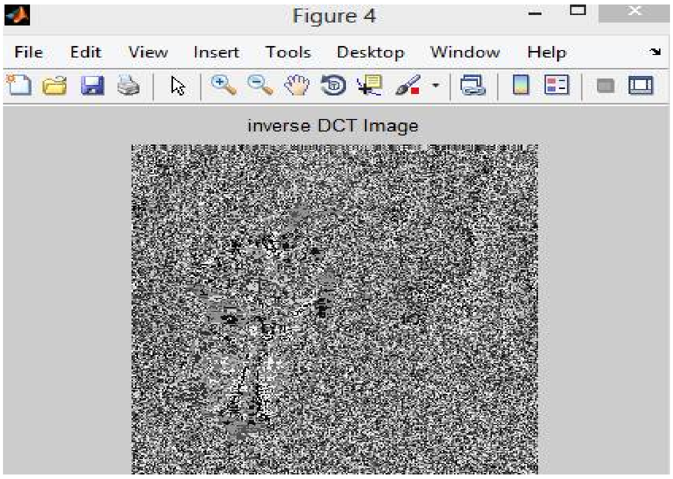

Figure 4.5 Steganographic image retrieval using IDCT

From the figure 4.5 it is obvious that retrieved image is same as that of image 4.2. And its PSNR with respect to steganographic image is 321.346, which are quite high and reasonable.

It can be concluded on the basis of PSNR that steganographic image is worse for the hacker, because the PSNR for image retrieval is poor, and it should be at least 30.

V. CONCLUSION

In this paper I propose a combination of the data hiding technique and cryptography scheme for high resolution image. Our intension is to provide proper protection on data during transmission. The process of data hiding (steganography) has been completed via bit swapping whose algorithm is described in earlier section. The resulted figures from the process of steganography regards PSNR have produced satisfactory results in the field of security with and without compression. The resulted figure of PSNR with and without compression are 8.4755, 4.8492. Its main advantage is that it is a blind scheme and its affect on image quality or coding efficiency is almost negligible. It is highly configurable, thus it may result in high data capacities. Finally, it can be easily extended, resulting in better robustness, better data security and higher embedding capacity.

REFERENCES

[1] Arvind Kumar and KM. Pooja "S*teganography- a data hiding technique*", international journal of computer applications (0975 – 8887) volume 9– no.7, November 2010.

[2] Adel Almohammad *"Steganography-Based Secret and Reliable Communications: Improving Steganographic Capacity and Imperceptibility"* A thesis submitted for the degree of Doctor of Philosophy, Department of Information Systems and Computing , Brunel University, August, 2010.

[3] Debnath Bhattacharyya, P. Das, S. Mukherjee, D. Ganguly, S.K. Bandyopadhyay, Tai-hoon Kim, *"A Secured Technique for Image Data Hiding"*, Communications in Computer and Information Science, Springer, June,
2009, Vol. 29, pp. 151-159.

[4] S.K. Bandyopadhyay, Debnath Bhattacharyya, D. Ganguly, S. Mukherjee and P. Das, "*A Tutorial Review on steganography"*, International Conference on Contemporary Computing (IC3-2008), Noida, India, August 7-9, 2008, pp. 105-114.

[5] Steganography and Steganalysis, J.R. Krenn, January 2004.

[6]Spyridon K. Kapotas, Eleni E. Varsaki and Athanassios N. Skodras, *"Data Hiding in H.264 Encoded Image Sequences"*, IEEE 9th Workshop on Multimedia Signal Processing, October 1-3, 2007, Crete, pp. 373-376 .

[7] A. Ker,"*Steganalysis of Embedding in Two Least-Significant Bits*", IEEE Trans. on Information Forensics and Security, vol. 2, no. 1, March 2007, pp. 46-54.