International Journal of Applied Engineering Research ISSN 0973-4562 Volume 12, Number 16 (2017) pp. 5565-5569

Data Hiding Using Fibonacci EDGE Based Steganography for Cloud Data

Davuluri Suneetha[1] and Dr. R.Kiran Kumar[2]

[1]Research Scholar, Computer Science
Krishna University, Machilipatnum, Andhra Pradesh, India.

Orcid: 0000-0001-5135-5272

[2]Assitant Professor, Computer Science, Krishna University
Machilipatnum, Andhra Pradesh, India.

Abstract

Cloud Computing is a powerful, flexible, cost efficient platform for providing consumer IT services over the Internet. However Cloud Computing has various levels of risk factors because most important information is outsourced by third party vendors, which means harder to maintain the level of security for data. Steganography is art of hiding information in an image. In this most of the techniques are based on the Least Significant Bit(LSB) bit ,but the hackers easily detect as it embed data sequentially in all pixels .Instead of embedding data sequentially some of the techniques choose randomly. A better approach for this chooses edge pixels for embedding data. So we propose novel technique to hide the data in the Fibonacci edge pixels of an image by extending previous edge based algorithms. This algorithm hides the data in the Fibonacci edge pixels of an image and thus ensures better security against attackers.

Keywords: Steganography, Edge Detection, Fibonacci, Encryption, Decryption, Stego Image

INTRODUCTION

Cloud Computing is a powerful, flexible, Cost efficient platform for providing consumer IT services over the Internet. Cloud Computing provides various services to the different sectors like IT, Medical etc and it is rapidly growing now days. However Cloud Computing has various levels of risk factors because most of the services are handled by third party vendors, which means some important information is also maintained by third party, harder to maintain the level of security.

Steganography comes from the Greek Words: Steganos-"Covered", Graphe-"Writing". The main goal of steganography is to provide security for most important data of a user by hiding it into an image and it protects data from attackers. Steganography is the art and science of writing hidden messages in such a way that no one can knows of the existing of the message. Due to growing need for security of data image steganography is so popular now a day. There are many steganography applications for digital image, including copyright protection, feature tagging, and secret communication [11,12]**.** Unfortunately the members of terrorist organizations are using steganography as a tool to attack against the western interests [13,14]. The traditional image Steganography algorithm is based upon the LSB bits but it can easily detect by attackers because the data is embedded in all pixels of the image. If the attacker identity at least one pixels location easily the attackers hack the original entire message. Instead of sequentially hiding the data in all pixels some of the algorithms are choose random pixels to hide the data, but it causes speckles in a image.

To overcome these problems we proposed a novel image steganography algorithm based on Fibonacci edge pixels for hiding secret message in the Fibonacci edges of the image. The Organization of the paper is as follows. In section 2 the related work is discusses. In section 3 our proposed method is described. Finally the results are presented in section 4.

RELATED WORK

The data can be visible in many basic formats like Audio, Video, Text, Image etc those are easily detect by the humans for hiding those the ultimate solution is steganography . There are different ways of Steganography.

Image Steganography:

In which we hide the data within a image so that there wiil not be visible to the human directly.

Audio Steganography:

Steganography can be applied to audio files. In which we hide the data within a audio file so that there will not be visible to the human directly.

Video Steganography:

In which we hide the data within a video file , the data is in compressed format but not visible the attacker directly.

Least Significant Bit Embedding:

In this image is converted into a gray scale image. In a gray scale image each bit is represented 8 pixels. The last bit in the pixel is called as least significant bit. After identifying the LSB bit of an pixel we can place the data. So many algorithms was proposed based on the LSB bits but attackers easily detect the hidden by identifying LSB bit of a given pixels. In this we have some more algorithms based on the LSB bit. In those we select not only a single LSB bit by combination of LSB and MSB bit we apply some more mathematical operation like bit OR ,bit XOR operation while inserting secret data either 1 or o.

Random Least Significant Embedding:

In this algorithm data is hidden randomly. In this image is converted into a gray scale image. In a gray scale image each bit is represented as 8 pixels. For inserting secret data within a image gray scale pixels are selected randomly.

The steganography by using existing algorithms can be easily detected by the attackers as they hide the data either in least significant bit or in randomly selected pixels. So we proposed a noval algorithm to hide the data within a image.

PROPOSED METHOD

In this proposed method we use all the Fibonacci edge pixels in an image. First the original image is converted into gray scale image. Then we identify edge pixels by using canny edge detection method. After obtaining the edge pixels we select Fibonacci edge pixels in a stego image.

Secondly read the original file which consist of original data and encrypt the data by using encryption algorithm and obtain the key, and then we hide the key in those Fibonacci edge pixels

At the receiver the stego object is again masked at the Fibonacci edge pixels. Then the canny edge detector is used to identify the edge pixels. We will get same edge pixels at the sender and receiver since we used the same image to calculate the edge pixels. Thus we identify where key is hidden, then decrypt the data based on key by using decryption algorithm.

ALGORITHM FOR INSERTION OF MESSAGE

Input: Input image

Input message

Output: Stego object

Step1: Start.

Step2: Import image using imread() function

Step3: Select image and convert into gray scale using formula rgb2gray.

Step4: Identify edge pixels from gray scale image using canny edge detection algorithm.

Step5: Read original text and encrypt original data and obtain key

Step6: Select Fibonacci pixels from edge based image

Step 7: hide the key into the pixels

Step 8: Obtain the stego object

Step 9: stop

ALGORITHM FOR RETRIEVAL OF MESSAGE

Input: stegoobject :

Output: Message

Step1: Start

Step2: Import stego image using imread() function

Step3: Identify edge pixels from gray scale image using canny edge detection algorithm.

Step4: Identify Fibonacci edge pixels from canny edge image

Step5: Read the hiding data from the image and decrypt original data obtain key

Step6: Decrypt original text based on key using decryption algorithm

Step 7: stop

RESULTS AND DISCUSSIONS

The Fibonacci edge based steganography is to embed secret data in the position of Fibonacci edge pixels, which meets the requirements of both in perception and robustness.

The Fibonacci edge based steganography includes Algorithms III.1 and III.2 for encoding and decoding process respectively. We have used different gray scale images for justifying the process.

Figure 1(a): Edge Image

Figure 1: Original image

Figure 1(a) Edge image

The Fig. 1 is Original image and Fig 1 (a) Edge image obtained after applying the Canny Edge detector

Figure 1(b): Fibonacci Edge Image

Figure 1(c): Stego image

The Proposed algorithm is applied on Fig 1(b).

Fig. 1(b) is the cover image and the Fig 1(c) is the stego image.

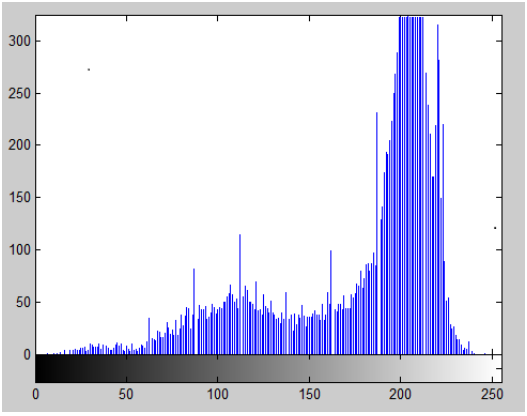

Figure 1(d): Histogram of the cover image and stego images

The Fig. 1(d) shows the histogram of the cover image and stego images of on a figure- Lena.

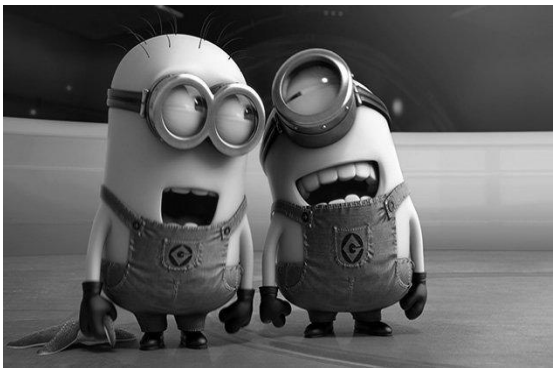

Figure 2: Cover Image

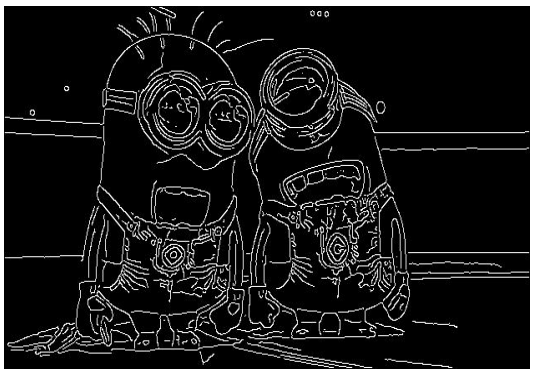

Figure 2 (a): Egde Image

The Fig. 2 is Original image and Fig 2 (a) edge image obtained after applying the Canny Edge detector

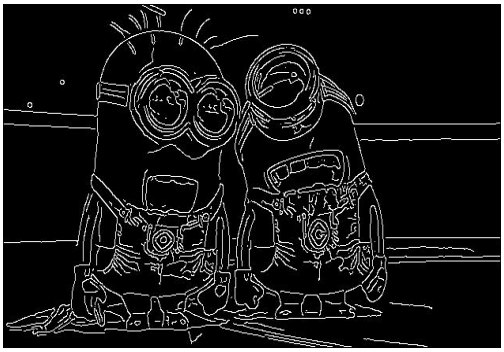

Figure 2(b): Fibonacci Edge image

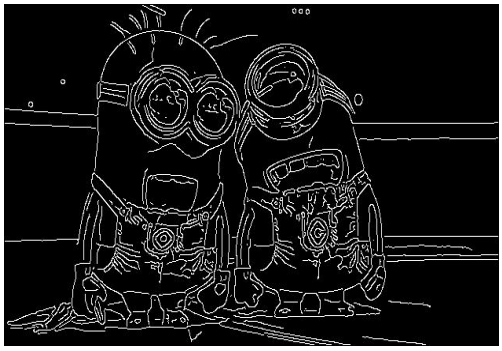

Figure 2 (c): Stego image

The Proposed algorithm is applied on Fig 2(b).Fig 2(b) is the cover image and the Fig 2(c) is the stego image.

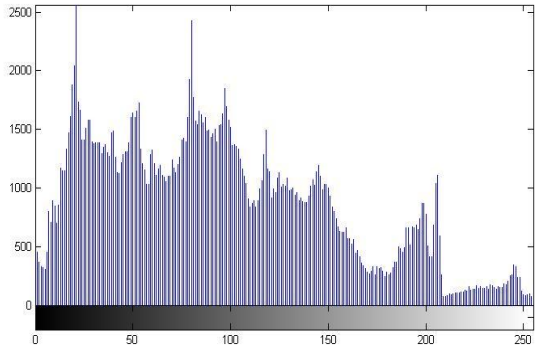

Figure 2(d)

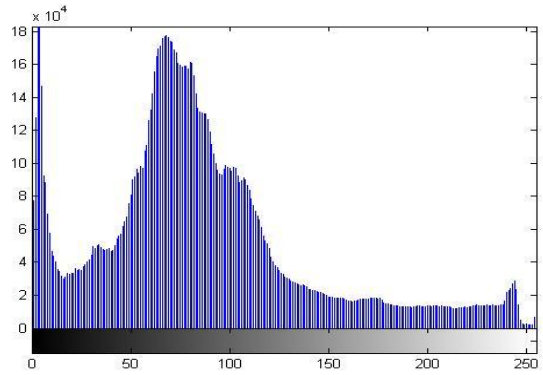

Figure 2(e)

The Fig2(d) and Fig 2 (e) shows the histogram of the cover image and stego images of on a figure- Mini.

CALCULATION OF (PEAK SIGNAL TO NOISE RATIO):

The PSNR square figures the pinnacle flag to commotion proportion, in decibels between two pictures. The higher the PSNR, the better the nature of the compacted or remade picture. We can calculate by using this formula.

$$PSNR=10\ log_{10}(MAX_i^2)/MSE$$

Where MAX_i is the maximum possible pixel value of the image when the pixels are represented using 8 bits per sample this is 255.

CALCULATION OF (MEAN SQUARE ERROR):

The Mean Square Error (MSE) and the Peak Signal to Noise Ratio (PSNR) are the two important measurement used to think about picture pressure quality. The MSE speaks to the total squared large difference between the compacted and the first picture. The lower the estimation of MSE, the lower the mistake.

$$MSE=\sum_{M,N}[I_1(m,n)-I_2(m,n)]^2/M*N$$

COMPARISON TABLE:

This Section contains the comparison between the previous work and the proposed calculated values of MSE and PSNR where it can be clearly seen that calculated values shows some significant decrement which suggests that the proposed approach is slightly better than the previous approach. The table was generated in the MATLAB tools which were also used for the calculation of MSE and PSNR values of the output image.

Table 1: Previous and Proposed MSE values

Cover Image	Previous MSE	Proposed MSE
Lena	0.0048	0.0046
Mini	0.0046	0.0043
Cameraman	0.0021	0.0020
Baboon	0.0004	0.0004

Table 2: Previous and Proposed PSNR values

Cover Image	Previous PSNR	Proposed PSNR
Lena	72.89	73.01
Mini	74.71	74.89
Cameraman	82.89	82.98
Baboon	82.03	83.01

CONCLUSION

In the Least Significant Bit embedding algorithm (LSB) and Random Least Significant Bit embedding algorithm (RLSB) an attacker can easily detect the presence of hidden image. To overcome these problems a new algorithm is proposed for hiding secret messages in the Fibonacci edges of the image. The algorithm hides data in Fibonacci edge pixel. The proposed algorithm is applicable to all kinds of images and can be used in covert communication, hiding secret information.

REFERENCES

[1] R. Chandramouli and N. Memon,"Analysis of LSB based Image Steganography", IEEE ICIP, pp. 1022-1022, Oct. 2001.

[2] R.J. Anderson, F.A.P. Petitcolas, “On The Limits of Steganography”, IEEE Journal of Selected Area in Communications, pp. 474-481, May 1998.

[3] N.F. Johnson, S. Jajodia, “Stag analysis: The Investigation of Hiding Information”, IEEE, pp. 113-116, 1998.

[4] H.Hastur,Mandelsteg,ftp://idea.sec.dsi.unimi.it/pub/security/crypt/code/

[5] K. Rabah, "Steganography- the Art of Hiding Data", Information Technology of Journal, 3(3), pp.245-269, 2004.

[6] http://fbi.edgesuite.net/libref/historic/famcases/dickinson/dickinso n.htm

[7] N.F.Johnson and S. Jajodia, "Exploring Steganography: Seeing the Unseen", Computer 31, pp.26-34, 1998.

[8] http://en.wikipedia.org/wiki/null_cipher

[9] F.A.P. Petitcolas, R.J. Anderson and M.G. Kuhn, “Information Hiding - A Survey", IEEE Proc., Special Issue on Protection of Multimedia Content, 87(7),pp.1062-1078, July 1999.

[10] http:/wetstonetech.com/f/stego/kessler.pdf

[11] N.F. Johnson, S. Jajodia, “Stag analysis: The Investigation of Hiding Information”, IEEE, pp. 113-116, 1998.

[12] D. Artz, “Digital Steganography: Hiding Data within Data”, IEEE Internet Computing, pp. 75-80, May-Jun 2001.

[13] D. Verton, "Expert Debate Biggest Network Security Threats", USA Today, 12 April, 2002.

[14] K. Maney, "Bin Laden's Messages could be Hiding in Plain Sight", USA Today 19 December, 2001.

[15] N.Provos,P.Honeyman,“Detecting Steganography Content on the Internet”. CITI Technical Report 01-11,oct-09,2001.

[16] N. Provos, “Probabilistic Methods for Improving Information Hiding”, CITI Technical Report 01-1, January 31,2001.

[17] N.F. Johnson &S. Jajodia, “Steganalysis of Images Created Using Current Steganography Software”, in Proceeding for the Second Information Hiding Workhop, Portland 0 regon, USA, April 1998,pp. 273-289.

[18] D. Artz, "Digital Steganography: Hiding Data within Data", IEEE Internet Computing, pp. 75-80, May-June 2001.