Academic Science

International Journal of Innovations & Advancement in Computer Science
IJIACS
ISSN 2347 – 8616
Volume 4, Special Issue
May 2015

Probabilistic approach for Hypothesis Testing: Steganography

Shreerang Hegde
B. Tech. Student
Computer Science & Engineering Department
Institute of Technology
Nirma University

Rupal Kapdi
Assistant Professor
Computer Science & Engineering Department
Institute of Technology
Nirma University

Abstract
Steganography detection is the process of detecting hidden messages in seemingly innocuous data. Messages can be hidden in almost all forms of data, like audio, video, images etc. There are different types of methods and tools via which messages can be hidden in data streams. There are also many methods via which we can detect whether a bit stream has undergone Steganography or not. In this paper probabilistic approach for Steganalysis (Steganography Detection) on images is performed where Least Significant Bit Flip method is used which involves manipulation of LSB of the pixel value of an image. This method is based on the probability of occurrence of the message given the entire image.

1. Introduction

Steganography is the technical term for covered text. It means hiding messages in anything and everything that can be represented in the form of stream of bits, be it audio or video, image etc. It is used for passing covert messages between two parties without any third party intercepting the message. A bit stream over which Steganography is used does not attract attention. One can only find the hidden message if one is looking for it. So there is a high chance that confidential or sensitive data may go undetected by the third party unless they are specifically looking for it. Steganography may hide the data from human eyes, but it may not necessarily hide the data from the computer. In their simplest forms anagrams can be considered to be an example of cryptography and writing messages in invisible ink is an example of Steganography.

Steganalysis is the process of determining whether a particular stream of bits has been subjected to Steganography or not. Since it is not definitive that a particular stream of bits contains a hidden message, steganalysis needs to be performed on the entire stream of bits in order to determine whether a hidden message is present or not. Robust methods hide messages over significant areas making them immune to attacks of compression, cropping and other image processing transformations which make use of original untampered image for extraction of certain hidden messages. Hiding information in media may degrade some of its properties. Vulnerabilities in Steganographic methods may be exploited in several ways like detecting, extracting, disabling or destroying information. Different Steganographic tools may use different methods and stegokeys. Patterns in images become visible after evaluating many images by comparing the original images and stegoimages and visible images are noted and the tools used in Steganography can be found out. Steganalysis is a procedure to find whether Steganography has been used on a certain piece of data or not [1].

There are so many methods available for stegnalisis. For eg. Non-blind stegnalysis is relatively easy method in which we have an original untouched image and the stegoimage. Then we compare the LSB of each and every pixel of the stegoimage with the respective pixel in the original untouched image. Whenever we encounter a bit mismatch between the two we extract the bit from

the stegoimage. We follow this procedure for each and every pixel in the images. In the end all the extracted bits are grouped into sets of 7 bits in order to get a ASCII value of a character. This process works well only when we have a cover image. If we don't have the cover image then this method fails. If we use blind stegnalysis where we do not have the original image for comparison then we need to extract each and every possible word from the stegoimage using LSBs of each pixel. We extract LSBs of 7 consecutive LSBs then determine whether they lie in the ASCII range of characters. If they do not, we extract 7 LSBs from the pixel after the first one used in the current pass. However there is major drawback in this method. Suppose the bit stream is as follows:- 1,0,0,1,0,0,1,0,0,0. The intended hidden message is 1,0,0,1,0,0,0 which starts from position 4. However 1001001 also lies in the ASCII range. Therefore 1001001 will be extracted and the intended hidden message will not be known. In this case the method fails. One way around this drawback or to reduce the probability of erroneous extraction of hidden message can be done by using Unicode characters which makes use of 16 bit character range. This method merely reduces the probability of erroneous extraction of hidden message since it requires 16 consecutive bits to be available in the required format. However the problem that we encountered in the ASCII method will still be present in this method though it will not be as severe.

2. Proposed Method

This paper deals with Steganography is done in digital images and the hidden message is assumed to be in text form with each character represented as its respective ASCII value. LSB Flip is a method by which Steganography can be implemented in images. In this method we alter the LSB (Least Significant Bit) of the pixels in the image. By flipping the LSB in certain pixels an encoded message can be hidden within a particular image. A message whose characters are stored in the form of ASCII values can be hidden in an image. One character is represented as 7 bits in ASCII form. So we need 7 LSBs from 7 different pixels for one character. The image consists of pixels having grayscale intensity values. Now if the message size is say 7 characters and image size is 256x256 then probabilistic approach says :

P(correctly identifying message) = embedded message/ all possible messages in image…… (1)

If we try to evaluate above equation just for the one data message of length 5 then total no of possible messages in a 256x256 grayscale digital image is 13107, which is approximately zero.
R code for proposed approach

The following code is for steganalysis when we have the keywords which may occur in the stegoimage.
For simplicity here pixel position from which data insertion has started is fixed.

```
img<-readJPEG('/Users/Public/Pictures/Sample Pictures/Tulips1.jpg')
z<-1

#Message to be hidden in image
h<-c(1,0,0,1,0,0,0,1,0,0,0,1,0,1,1,0,0,1,1,0,0,1,0,0,1,1,0,0,1,0,0,1,1,1,1)
l<-c(1,0,0,1,0,0,0,1,0,0,1,0,0,1)
#Hiding message in image
for(x in 1:35)
{
```

Academic Science

International Journal of Innovations & Advancement in Computer Science
IJIACS
ISSN 2347 – 8616
Volume 4, Special Issue
May 2015

```
  a<-img[550,x+550,3]*255
  i<-a%%2
    if(h[x]==1)
  {  if(i==0)
    {
     img[550,x+550,3]<-(img[550,x+550,3]*255+1)/255
    }
  }
    if(h[x]==0)
  {  if(i==1)
    {
     img[550,x+550,3]<-(img[550,x+550,3]*255-1)/255
    }
  }
}
#Extracting Message from image
flag<-0
cnt<-0
o<-c(0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0)
i<-1
j<-0
for(x in 1:768)
{
  for(y in 0:989)
  {
k<-1
     for(z in 1:35)
    {
     if(((img[x,y+z,3]*255)%%2)==h[z])
      {
        o[k]=img[x,y+z,3]*255
        k<-k+1
      }
     if(k==36)
       break
    }
    if(k==36)
     break
  }
  if(k==36)
    break
}
```

3. Conclusion

In this paper Steganography is done when the text message size is known and it is inserted at random position in the image. Depending on the text size and image size it is approximately not possible to detect the message from the image which makes the stegnalysis difficult.

International Journal of Innovations & Advancement in Computer Science
IJIACS
ISSN 2347 – 8616
Volume 4, Special Issue
May 2015

4. Future Work

In future this method can be extended where text message which is to be hidden is shuffled and then instead of putting it in sequence we apply some random sequence to insert this method.

5. References

1. Neil F. Johnson and Sushil Jajodia, "Steganalysis: The Investigation of Hidden Information", 1998 IEEE Information Technology Conference, Syracuse, New York, USA.
2. Dr. Diwedi Samidha and Dipesh Agrawa1, "Random Image Steganography in Spatial Domain", 978-1-4673-5301-4/13 2013 IEEE
3. Natarajan Meghanathan and Lopamudra Nayak, "Stegnalysis Algorithms for detecting the hidden information in Image, Audio and Video cover Media", January 2010 International Journal of Network Security & Its Application (IJNSA), Vol.2, No.1
4. Jeremiah J. Harmsena and William A. Pearlmana, "Steganalysis of additive noise modelable information hiding".
5. Jessica Fridrich, Miroslav Goljan, Dorin Hogea, "Steganalysis of JPEG Images: Breaking the F5 Algorithm".