Project Report on

Secure Communication of Information Using Steganography

Submitted in the 3rd Semester of

BACHELOR OF TECHNOLOGY

(COMPUTER SCIENCE AND ENGINEERING)

Of

INDIAN INSTITUTE OF INFORMATION TECHNOLOGY, KALYANI

By

AKARSH SOMANI (0000162)

GAURAV MISRA (0000172)

Under the guidance of

Dr. IMON MUKHERJEE

(Assistant Professor, IIIT-KLY)

NOV, 2017

Abstract

Steganography is the art of hiding the fact that communication is taking place, by hiding information in other information. Many different carrier file formats can be used, but digital images are the most popular because of their frequency on the internet. For hiding secret information in images, there exists a large variety of steganography techniques some are more complex than others and all of them have respective strong and weak points. Different applications may require absolute invisibility of the secret information, while others require a large secret message to be hidden. This project report intends to give an overview of image steganography, its uses and techniques. It also attempts to identify the requirements of a good steganography algorithm and briefly reflects on which Steganographic techniques are more suitable for which applications..

Steganography is the practice of hiding private or sensitive information within something that appears to be nothing out to the usual. Steganography is often confused with cryptology because the two are similar in the way that they both are used to protect important information. The difference between two is that steganography involves hiding information so it appears that no information is hidden at all. If a person or persons views the object that the information is hidden inside of he or she will have no idea that there is any hidden information, therefore the person will not attempt to decrypt the information.

What steganography essentially does is exploit human perception, human senses are not trained to look for files that have information inside of them, although this software is available that can do what is called Steganography.

INTRODUCTION TO STEGANOGRAPHY

Steganography is the art of passing the information without letting others know about the existence of message. The goal of steganography is to avoid suspicion to the transmission of hidden message therefore making it highly secured. It can be viewed as akin to cryptography. Both have been used to protect information in the past. Cryptography technique ‘scramble’ message making it difficult to understand for other person. While Steganography ‘embed’ a message to hide its existence and make it seem ‘invisible’ thus concealing the fact that a message is being sent altogether. It is not intended to replace cryptography but supplement it. One of the most common methods is Least Significant Bit (LSB) Algorithm. This deals with the embedding of message along with the image file where each pixel is of size 3 bytes. Each and every bit of the message is taken and is embedded along with the bytes of the image file such that, its inclusion does not make a perceivable change in the message embedded file. The simplicity of the algorithm decides the level of security of the massage.

Here considering the confidentiality of the massage we can do these things to keep it secure-

1.) Use of digital watermarking.
2.) Use of Cryptography (we have developed our own algorithm for that).
3.) Ability to send more than one massage files in the same image file.
4.) Password Authentication.

Origin of Steganography

Steganographic techniques have been used for ages and they date back to ancient Greece. The aim of steganographic communication back then and now, in the modern applications, is the same: to hide secret data in an innocently looking cover and send it to the proper recipient who is aware of this hidden communication.

The first recorded use of steganography was in 1499 by Johannes Trithemius in his book Steganographia, which was called a magic then. Some methods used then were - hiding information on the scalp by writing on it and then covering it by hairs, writing on a paper by lemon juice which is invisible unless it is brought near heat, Hiding messages within wax tablets etc.

The form of cover had been developed since then. In the ancient time they used physical steganography like human skin, game etc. The increasing literacy among people had brought some more methods like text itself. The world wars had accelerated the development of steganography by introducing a new carrier – the electromagnetic waves. And nowadays the most popular carriers include digital images, audio and video files and communication protocols. But the general principles remain unchanged.

Existing Systems:

The existing system algorithm in steganography is LSB algorithm. This means Least Significant Bit algorithm. This steganography entered into the computer technologies during the hours of September 11 incidents by osama bin laden. But before that it was resident and was used for secured communication of messages outside computer technology.

Back Ground

Steganographic Software is new and very effective. Such software enables information to be hidden on graphic, sound and apparently "blank" media.

In the computer, an image is an array of numbers that represent light intensities at various point called pixels in the image. A common image size is 640 by 480 and 256 colours or 8 bits per pixel. Such an image could contain about 300 kilobits of data.

There are usually two types of files used when embedding data into an image. The innocent looking image, which will hold the hidden information, is a "container". A "message" is information to be hidden. A message may be a plain text, cipher text, other images or any other things that can be embedded in the LSB of an image.

For example:

Suppose we have a 24-bit image 1025 x 768 (this is a common resolution for satellite images, electronic photographs and other high resolution graphics). This may produce a file over 2 megabytes in size. All colour variations are derived from three primary colours – Red, Green and Blue. Each primary colour is represented by one byte. 24-bit images use 3 bytes per pixel. If information is stored in the LSB of each byte, 3 bits can be stored in each pixel. The "container" image will look identical to the human eye, even if viewing the picture side by side with the original. Unfortunately, 24-bit images are uncommon and quite large. They would draw attention to themselves when being transmitted across a network. Compression would be beneficial if not necessary to transmit such a file. But file compression may interfere with the storage of the information.

LSB Algorithm

In LSB algorithm, the message bit is taken from the message byte and then that particular bit will be embedded inside the LSB of an image or video or audio file. This is done because:

1.) The message embedded in the LSB of an image file will not draw the suspicion of the hacker as the minute difference that would be made in the pixel value of the image file will not be perceived by the normal naked eyes.
2.) The message that will be embedded in the LSB of an audio file will not create suspicion to the hacker as that change would not be perceived by human ear.
3.) The same concept works out even with video file.
4.) This same algorithm can be used for digital watermarking.

Implementing Steganography in LSB:

Steganography can be implemented using LSB algorithm in two ways –

1.) Using keys
2.) Without using keys

Explanation:

Now let us see a small example of how the message will be stored. Consider 8 bytes of a cover file (say bitmap image). Let 8 bytes be:

11111010

10101010

11110000

01010110

10010010

01101110

10000100

11000000

Consider a message byte **11111111**. We should embed all these 8 1s into the image file. Since we are going to use the LSB algorithm, we need 8 bytes from cover file to embed a byte of message. This is done by, covering each and every bit of the message file by the LSB of the cover file.

The final answer would be

11111011

10101011

11110001

01010111

10010011

01101111

10000101

11000001

The 1s with underline denotes the message bit embedded in the cover file.

In case of LSB **without key**, it is nothing but embedding all the message bits in the successive bytes.

The file size will be embedded as a header in the embedding process itself by the sender to help the receiver while decoding thereby not wasting time in decoding the bytes that does not have any message embedded. Another constraint is that **size of the message file should not be greater than 1/8th of the size of the cover file.**

In case of **LSB with key**, we add an extra key such that based on the key value; the number of bytes will be traversed. For example after embedding in the LSB, the key value will be checked and if the key value is n, after n number of bytes the bit will be embedded in it. This type of embedding can be done in image file and it was once done by using the digital message that is being passed in **ISDN phone lines**. It was hiding message in the noise of ISDN telephone.

Stego key generated in the sender side should be generated in the receiver side to retrieve the message from the cover bytes. Else it would not be possible for the receiver to decode at the same time, the number of bytes needed in this case cannot be determined during the algorithm design time. It is totally dependent on the stego key and message file size.

Choosing Cover File:

The main thing that should considered in choosing an image file as a cover file is the compression method. There are two types of compression methods.

Lossless Compression (e.g. GIF, BMP)

- Original information remains intact.
- Original message can be reconstructed exactly

Lossy Compression (e.g. JPEG)

- Yields very good compression
- May not maintain the integrity of the original image

A 24-bit Bitmap Image:

a) A 24 bit bitmap image is made up of pixels of size 24 bits.
b) The pixel has 8 bits of Red, Green and Blue values respectively.
c) So the colour variations are derived from these 3 primary colours.
d) Information is stored in the LSB of 3 bytes.

Drawbacks of the Existing System:

Though LSB has the advantages like easy implementation, it also has its unavoidable drawbacks. Once if the hacker is aware of the presence of message in the cover file, he will first use the LSB algorithm and so it gives very less security to the message. More over as the name refers only LSB of the cover byte is used and other bits are not used as it may create changes in the cover file and it would draw suspicion and tends to easy steganalysis.

What we have done:

As there is no proper security promise in the LSB algorithm, we have developed a new Algorithm to avoid such threats. It includes –

1.) Inability to hack the content immediately
2.) First Encrypt and then embed
3.) Use of stego keys

The main concepts of Network Security is

1. The data should be sent from a recognized sender.
2. The data should reach the legitimate receiver.
3. Data Integrity.

In order to fulfil these requirements, we have implemented following things in our project:

1. Use of Stego Keys – “to check for legitimate receiver”
2. Cryptography and Steganography – “Data Integrity”

We have used two modules here:

The **encrypt module** is used to hide information into the image; no one can see that information or file. This module requires any type of image and message and gives the only one image file in destination. The **decrypt module** is used to get the hidden information in an image file. It take the image file as an output, and give two file at destination folder, one is the same image file and another is the message file that is hidden it that.

Our Algorithm:

1.) User will enter the message.

2.) We will encrypt the message by using our own method (We will first reverse the message and then exchange 1s by 0s and vice versa).

3.) User will inter a sequence of keys consisting of numbers between 0-7 (e.g. “62012646746”) and length of string will be the same as the length of message in bits.

4.) Now we will replace the LSB of pixel such that absolute difference in corresponding key bit and the LSB will be the message bit.

5.) Now in order to access the information from the image Receiver will need to get keys, which will be given by Sender.

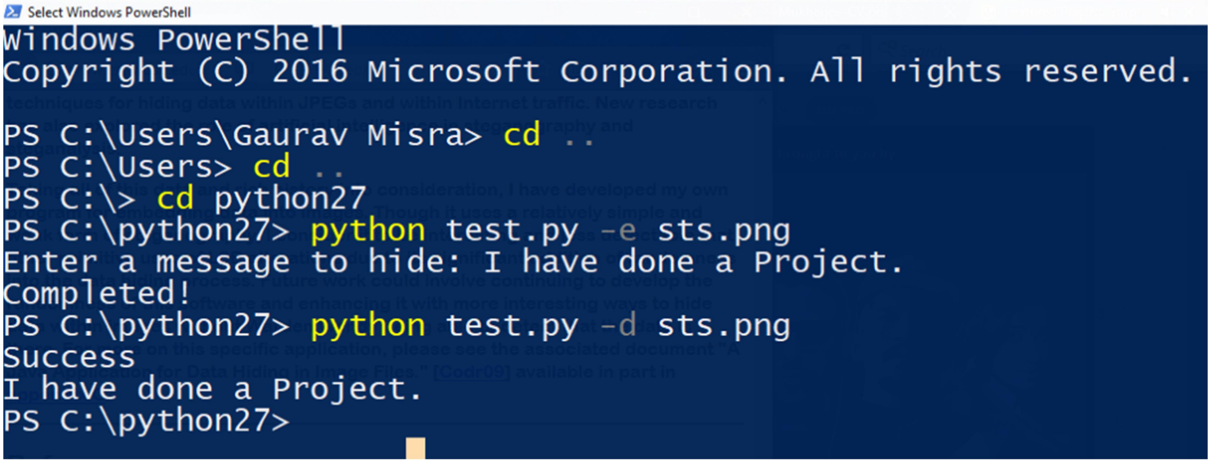
Our Code:

```
#Hiding Text in an image

from PIL import Image

import binascii

import optparse

def rgb2hex(r, g, b):

    return '#{:02x}{:02x}{:02x}'.format(r, g, b)

def hex2rgb(hexcode):

    return tuple(map(ord, hexcode[1:].decode('hex')))

def str2bin(message):

    binary = bin(int(binascii.hexlify(message), 16))

    return binary[2:]

def bin2str(binary):

    message = binascii.unhexlify('%x' % (int('0b'+binary,2)))

    return message

def encode(hexcode, digit):

    if hexcode[-1] in ('0','1', '2', '3', '4', '5'):

        hexcode = hexcode[:-1] + digit

        return hexcode

    else:

        return None
```

```
def decode(hexcode):
    if hexcode[-1] in ('0', '1'):
        return hexcode[-1]
    else:
        return None
def hide(filename, message):
    img = Image.open(filename)
    binary = str2bin(message) + '1111111111111110'
    if img.mode in ('RGBA'):
        img = img.convert('RGBA')
        datas = img.getdata()

        newData = []
        digit = 0
        temp = ''
        for item in datas:
            if (digit < len(binary)):
                newpix = encode(rgb2hex(item[0],item[1],item[2]),binary[digit])
                if newpix == None:
                    newData.append(item)
                else:
                    r, g, b = hex2rgb(newpix)
                    newData.append((r,g,b,255))
                    digit += 1
                else:
                    newData.append(item)
            img.putdata(newData)
            img.save(filename, "PNG")
            return "Completed!"

    return "Incorrect Image Mode, Couldn't Hide"
def retr(filename):
    img = Image.open(filename)
    binary = ''

    if img.mode in ('RGBA'):
        img = img.convert('RGBA')
        datas = img.getdata()

        for item in datas:
            digit = decode(rgb2hex(item[0],item[1],item[2]))
            if digit == None:
                pass
            else:
                binary = binary + digit
                if (binary[-16:] == '1111111111111110'):
```

```
                print "Success"
                return bin2str(binary[:-16])

        return bin2str(binary)
    return "Incorrect Image Mode, Couldn't Retrieve"
def Main():
    parser = optparse.OptionParser('usage %prog '+\
        '-e/-d <target file>')
    parser.add_option('-e', dest='hide', type='string', \
        help='target picture path to hide text')
    parser.add_option('-d', dest='retr', type='string', \
        help='target picture path to retrieve text')
(options, args) = parser.parse_args()
    if (options.hide != None):
        text = raw_input("Enter a message to hide: ")
        print hide(options.hide, text)
    elif (options.retr != None):
            print retr(options.retr)
    else:
        print parser.usage
        exit(0)
if __name__ == '__main__':
    Main()
```

Expected Output:



```
Windows PowerShell
Copyright (C) 2016 Microsoft Corporation. All rights reserved.

PS C:\Users\Gaurav Misra> cd ..
PS C:\Users> cd ..
PS C:\> cd python27
PS C:\python27> python test.py -e sts.png
Enter a message to hide: I have done a Project.
Completed!
PS C:\python27> python test.py -d sts.png
Success
I have done a Project.
PS C:\python27>
```

Future Amendments

We can do many improvements in future in order to make the Communication of Information more and more secure by using new and improved methods, some of them are following:

1.) Use of Watermarking
2.) Use of Advanced Concepts of Steganography
3.) Use of More Convenient Algorithm

Conclusion

In this Project, We have learnt many things about Steganography and also implemented some of them in our Program. We have proposed our own Algorithm that is more secure than LSB Algorithm, making the communication more secure. So we have presented an Overview of steganography starting with definitions and Basic Principals and proceeding through cover Media types, specific techniques and finally our own Algorithm of Steganography.

Many forms of covers exist for hiding messages, from image, video, and sound to text to IP packets. There are many specific techniques for embedding data within these various mediums, and each has its own strengths and weaknesses. Some, such as LSB encoding, are considered especially weak, whereas transform domain and feature modification techniques may be slightly stronger. Other cutting edge developments are emerging to create new methods to both hide and uncover data, and even to completely rethink the way steganography is used. Researchers have contemplated developing steganography that hides messages from computers rather than hiding them from humans, and have developed more advanced techniques for hiding data within JPEGs and within Internet traffic. New research has also explored the role of artificial intelligence in steganography and steganalysis.

Taking all of this information into consideration, we have developed our own compact Algorithm for embedding data into images. It can be further developed to make a strong, almost impossible to detect and promising Program.