Project Report

Steganography

Rijul (14BCE0914)

Abstract

Every single byte of information we send online can be kept track of /intercepted by third parties and used for malicious purposes. Therefore hiding this information becomes vital. Steganography is an ancient technique which unlike encryption/cryptology conceals the very existence of the message/file. To every packet tracer and MITM attacker out there, our traffic appears normal whereas only the sender and receiver can encode/decode the image/video.

In this project our aim is going to hide our “secret” message media here string in a base image by bit manipulation. A console application in python will be created to simulate the advantages of steganography over encryption.

Introduction

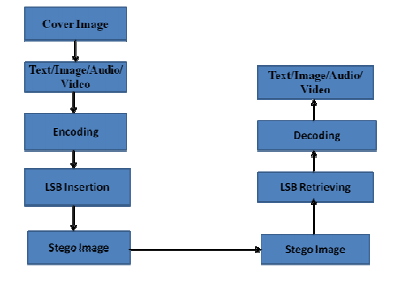
The basic structure of Steganography is made up of three components:

1. The Carrier image
2. The Message
3. The Key

The carrier can be a painting, or a digital image. It is the object that will carry the hidden message. A key is used to decode/decipher/discover the hidden image. This key or method of hiding is known only to sending and receiving parties. If any person knows how the media is being encoded then he/she can reverse engineer the method and use it for his own algorithm or worse discover a method to decrypt the messages hidden in the media

In this simulation I am going to be using python in order to make use of simple standard libraries like PIL to open and access bits in images in a much simpler and abstract way. If the image is of format PNG I ping the user to enter a message and then break the image into bits, hide my message, flag the end, and send it. On the other end I do the reverse and receiver gets the message.

Literature Survey and Methodology

Steganography has been a way to hide messages and other media since ancient times. The diagram given alongside is the basic model that exists for LSB or Least Significant Bit Insertion method to hide messages in digital media

In this method LSB of a bit is replaced with bits from our message. This technique works great as to the human eye the steganography (final encrypted image) image will look identical to the carrier image. To a computer an image file is simply a file that show a different colors of intensities of light on different areas of an image.

For example, our aim is to hide the character ‘A’ in our carrier image. A sample raster (different from random/vector image) data for 3 pixels (9 bytes) may be:

00100111 / 11101001 / 11001000

00100111 / 11001000 / 11101001

11001000 / 00100111 / 11101011

Inserting the binary value of ‘A’ (1000001):

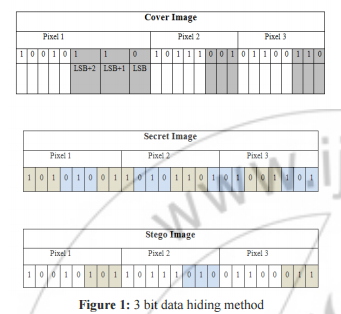
0010011**1** / 1110100**0** / 1100100**0**

0010011**0** / 1100100**0** / 1110100**0**

1100100**0** / 0010011**1** / 11101011

This is one of the most basic and most vulnerable method that is out there as if a MITM knows that information is being transferred via steganography method then he/she can just extract the LSB bits and retrieve the hidden image/message. The trick lies in finding a novel and safe way to embed the secret bits of the pixels/message characters in the cover image/audio/video.

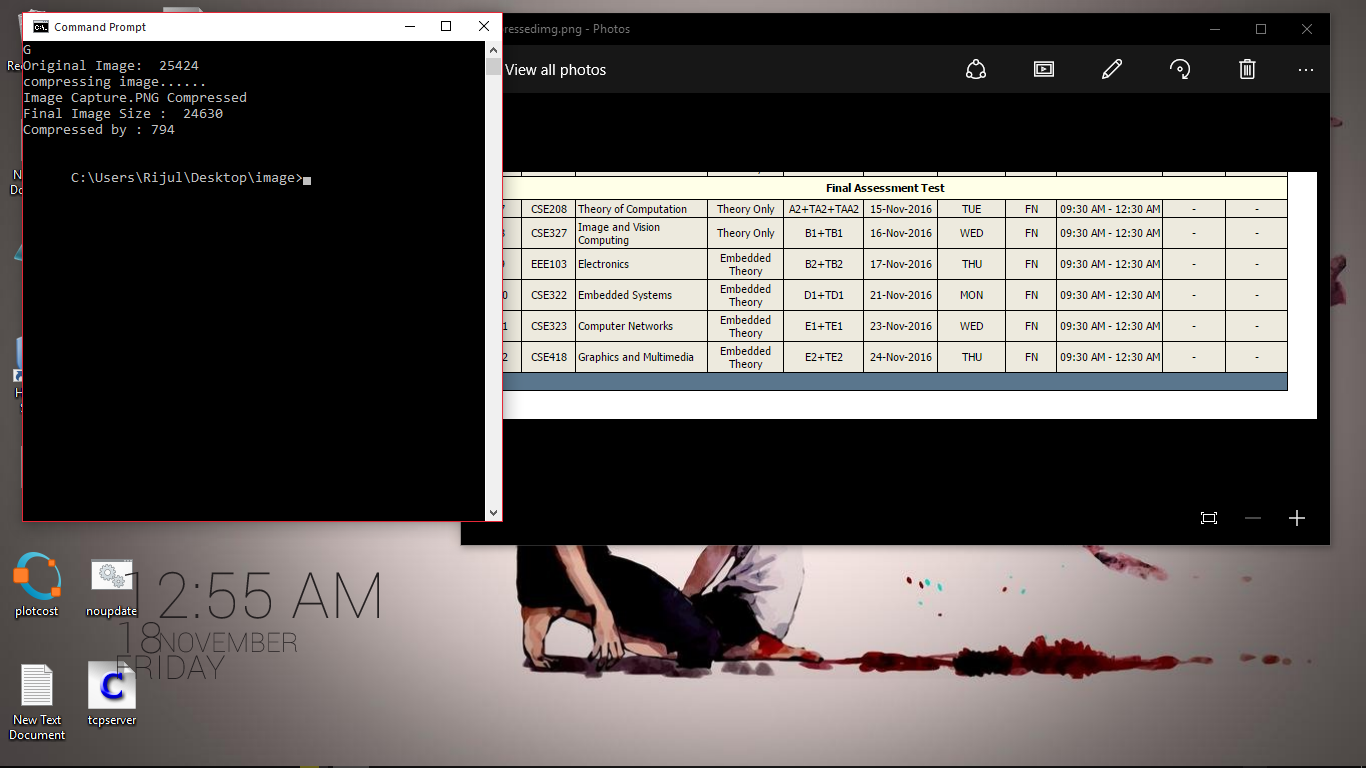
One of the newer and novel methods that I came across in one of the reference papers is discussed below

 A newer method in which it was suggested that using 3 bits from the secret image and embedding onto the cover image can be a preferred method. To improve it further a randomization technique was proposed. (Paper ref no 3 in references)RC4 algorithm (used in WIFI 802.11bgn for WEP/WPA2 protocols) can be used to find random locations in the cover image to hide our secret bits. This algorithm is used as it overcomes the sequence mapping problem (image from paper ref no 3 )

making the process more secure and also RC4 has an easy to implement key Scheduling algorithm. Even if an attacker identifies the existence of hidden information in a cover, it is difficult for him to recover it because the bits are embedded in a random order. The 3 bit data hiding method hides all the secret image bits at the same time offers improvement in information carrying capacity and allows regenerating the secret image at exact quality at the receiver

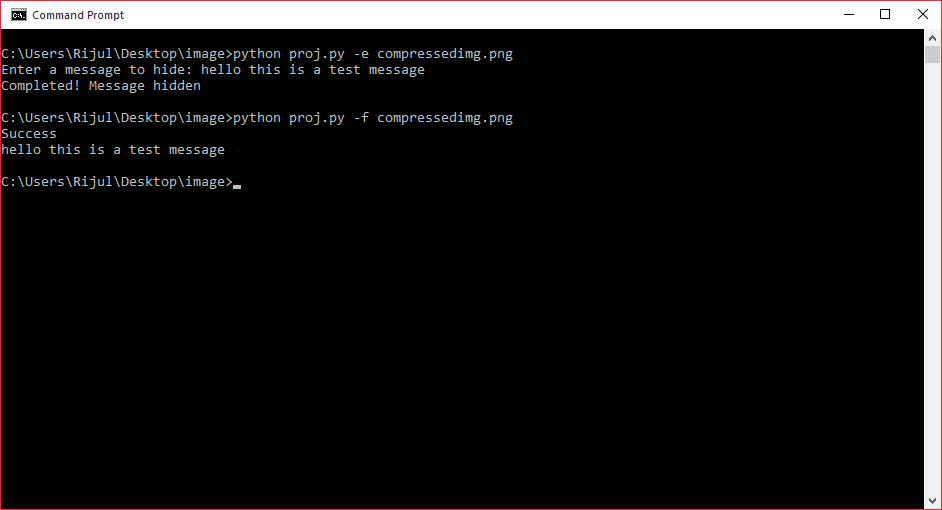
Results and Discussion

Output Screenshot 1



First I try to compress the image so that any malicious attacker who might judge the image by size isn’t aware of the message. Console on the left shows “-d ” option with image being given which compresses my image by about 800 bytes and stores as “compressedimg.png” which is the image in the right

Output Screenshot two



The console application is able to verify the “.png” format of the “compressedimg.png” and hide the user input message “hello this a test message” in the blue channel of the pixels having values in range 0-5 and even able to retrieve it character by character with zero noise.

Conclusion and Future Work

The image was successfully used to hide the message and the user on the ether side also received this message. To every packet tracer and MITM attacker out there, our traffic appears normal whereas only the sender and receiver can encode/decode the image/video.

Further the same method can be applied to handle various other media like audio and video and maybe even 3d graphic and PSDs too. They can have way more space and capabilities and may even hide some frames videos like gifs and small animations. Also various other algorithms can be implemented kike RC4 and end to end encryption to further the way and make the message more secure by making it completely gibberish even though its hidden in an incomprehensible image

References

1. Shilpa Gupta, Geeta Gujral and Neha Aggarwal, “Enhanced Least Significant Bit algorithm For Image Steganography”, IJCEM International Journal of Computational Engineering & Management, Vol. 15 Issue 4, July 2012 ISSN (Online): 2230- 7893, [www.IJCEM.org](http://www.IJCEM.org)
2. . H. Yang, X. Sun, G. Sun. “A High-Capacity Image Data Hiding Scheme Using Adaptive LSB Substitution”. Journal: Radio engineering Year: vol. 18, 4Pages/record No.: 509- 516, 2009.
3. Hiding Image in Image Using LSB Insertion Method with Improved Security and Quality Mekha Jose School of Computer Sciences, M G University, Kottayam Dst, Kerala, India
4. Other references include google searches and Wikipedia visits