



LEAST SIGNIFICANT BITS IN ELECTRONIC COMMUNICATION

Computer Science, Cyber Security and Digital Forensics'



DECEMBER 13, 2017

School of Informatics
Department of Information and Engineering
Institute of Technology, Blanchardstown
Dublin 15

Contents

Introduction	3
What is Steganography?	3
How does Steganography work?	3
History of Steganography.	3
Cryptography	4
Steganography	4
The difference between Steganography and Cryptography.....	5
Least Significant Bits (LSB)	5
Why do we use the (LSB)?	6
How do we use the (LSB)?	6
What if we want to hide an image?	7
The Evolution of Steganography.....	7
Ancient Greece.....	7
Roman times	7
Linguistic Steganography	8
Null Cipher Message sent from the us in WW1	8
World War 1.....	8
Spread Spectrum technologies	8
Jargon Code.....	9
Microdots.....	9
Subliminal Channels.....	9
Steganography in written text	9
Type spacing and offsetting	9
Digital Steganography	10
Network Steganography	10
How is Steganography on an Image.....	10
The basic function of Steganography coding:.....	10
Import a Library	11
First, libraries are imported. The PIL import Image is a Python Import Library to allow us to use images. The binascii is the binary library and the optparse is required.	11
Explaining Decoding and Encoding Functions:.....	11
A function is created to convert RGB to hex, and hex to RGB. The RGB is the colours of the image red green and blue. A function is created to convert string to binary and binary back to string. The encode function takes the first 0-5 blue and turns them to hexadecimal. The encode function is a reverse of the decode.	11
Encoding:.....	12

Retrieve Data Function:	13
Main Method:	13
The transformation of Steganography.....	15
Tools for Steganography	15
Conclusion.....	15
References	15
Gary C. Kessler	15
Security and Cryptography.....	16
Affiliated Faculty: James Aspnes, Joan Feigenbaum, Mike Fischer, Zhong Shao.	16
University of Yale	16
Gary C. Kessler February 2004 (updated February 2015).....	16
Websites visited	16
Watched videos on Steganography	17
https://www.youtube.com/watch?v=z_ypj5q5fzE – Principles of Steganography.....	17
https://www.youtube.com/watch?v=osNWSGsFOvA -	17
Network Steganography and Anomaly Detection	17

Introduction

Steganography is an art that has been around since 440B.C. It was used to transfer messages in secret between two parties. This paper will inform you of what steganography is and how it works. It also explains the history, evolution and development over the years and explain how Steganography has transformed in to computer technology using the **Less Significant Bits**.

What is Steganography?

Steganography is the process of hiding a secret message within a larger one in such a way that someone cannot know the presence and content of the hidden message. The purpose of Steganography is to maintain secret communication between two parties.

(Gary C. Kessler)

How does Steganography work?

Steganography is sometimes used when encryption is not permitted, or commonly used to supplement encryption. An encryption file may still hide information by embedding messages within other seemingly harmless messages. Steganography works by replacing bits of useless data in regular computer files such as graphics, sound text, HTML, or even floppy disks with bits that are different, invisible information.

(Gary C. Kessler)

History of Steganography.

Steganography technique has been dated back to ancient Greece. Its aim was to communicate back then and now.

In modern application, its uses are the same, to hide secret data in an innocently looking cover and send it to the proper recipient who is aware of the information hiding procedure. In an ideal situation the existence of hidden communication cannot be detected by a third party.

What distinguishes historical steganographic methods from the modern ones of today is, that only the form of the cover (carrier) for secret data has changed.

Historic methods were physical steganography on human skin or games. Further advances in hiding communication based on the use of more complex covers within the aid of ordinary objects, whose orientation was assigned meaning. This is how sonagram was introduced. The popularisation of the written word and the increasing literacy among people had brought about methods which utilised text as carriers. The word wars have accelerated the development of Steganography by introducing a new carrier, the electromagnetic waves. Presently most popular carriers including digital images, audio and video files and communication protocols. The letter may apply to network protocols as well as any other communication protocols. The way people communicate evolved over ages and so did steganographic methods. At the same time, the general principles remained unchanged.

(<http://stegano.net/tutorial/steg-history.html>)

Cryptography

Cryptography comes from the Greek term “KYPTROS” which mean hidden and “GRAPHEIN” meaning writing. Cryptography is a method of encoding a secret writing into a shadow image called shares, where each participant receives one share. It involves creating written or generated codes that allows information to be kept secret. Cryptography converts data in to an unreadable format for an unaware or unauthorized user. It is allowed be transmitted without anyone decoding it back into a readable format.

The information cannot be read without a key to decrypt it, so the information will keep its integrity while it is transmitted and while its stored.

(James Aspnes)

Steganography

Steganography comes from the Greek term “στεγανός/STEGANOS” which means covered or hidden written and is dated back to 440 B.C

The purpose of Steganography is to communication with a hidden message so that a third party doesn't know it exists.

Technical Steganography uses scientific methods to hide a message, such as the use of invisible ink or microdots and other size-reduction methods.

Semagrams hide information using symbols and signs. A visual semagram uses innocent looking or everyday physical objects to convey a message, such as doodles or the positioning of the items on a desk or Website. A text semagram hides a message by modifying the appearance of the carrier text, such as subtle changes in font size or type, adding extra spaces, or different flourishes in letters or handwritten text.

Open codes hide messages in a legitimate carrier message in ways that are not obvious to an unsuspecting observer. The carrier message is sometimes called overt communication whereas the hidden message is the covert communication.

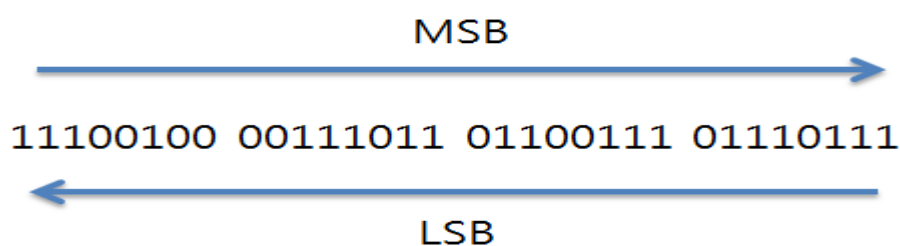
(Gary C. Kessler 2004)

The difference between Steganography and Cryptography

The difference between Steganography and cryptography is that cryptography, one can tell that a message has been encrypted, but cannot decode the message without the proper key.

In Steganography the message itself may not be difficult to decode, but most people would not detect the presence of it.

Least Significant Bits (LSB)



The Least Significant Bits (LSB) is the lowest bit in a series of binary numbers. The least significant bit is located at the far right of the binary number.

Example: 1001 110**1**

The last number **1** in red is the least significant bit.

The least Significant bits are commonly used in Steganography. Its advantage is it's easy to use. Its disadvantage is its more susceptible from attacks. Steganography is a way of concealing information within another piece of information.

LSB is the right most part of a binary sequence.

Most Significant Bit is on the left of the binary sequence. The number 1 in blue.

MSB

LSB

1001 1110

usually up to the 3rd bit can be used without distorting the message.

(Weissstein, Eric W.)

Why do we use the (LSB)?

Human observers will be unable to distinguish between the original image and the encoded image.

The pixels of the encoded image will be, at most, 1 value separated from the original

As you begin to encode using higher bits, severe degradation of the cover image occurs, defeating the purpose of encoding.

(Weissstein, Eric W.)

How do we use the (LSB)?

When converting an analogue image to a digital format we can choose 3 different ways of representing colours: 24-bit colour, 8-bit colour or 8-bit grey scale. The LSB insertion modifies the LSB's of each colour in a 24-bit, 8 bits for 8-bit images.

On every 8 bits we can infer 1 LSB insertion usually has a 50% chance of changing a LSB, adding very little noise to the picture.

On a 24-bit image we can modify up to the third LSB without being visible. The 8-bit images have much less space where to choose colours. It's usually only possible to change only the LSB without being detected.

For a 24-bit picture insertion can use 3 bits per pixel since every pixel has 24 bits. So, we can embed 3 bits every 8 bits of an image.

(Weisstein, Eric W.)

What if we want to hide an image?

An image is made up of individual pixels, which are 8 bits unsigned integers. Same principle can be applied to hide an image within a larger image.

Some restrictions

Cover images must be greater than 8 times the size of the hidden image.

The Evolution of Steganography

Steganography had a basic principle, there was a sender and a receiver. A hidden message of data was embedded into a carrier invisible to a third-party observer. Only receiver would be aware of the procedure to retrieve the hidden data from the carrier.

Ancient Greece

Harpagus wanted to send a message to Cyrus, but the roads were guarded, so he used a hare to conceal the message in its stomach. He used a servant dressed as a huntsman to carry the hare and told the receiver to cut open the sewed animal.

Histiaios wanted to inform Aristagoras that he should revolt. He shaved a servant's head and had it tattooed. When the hair had grown back, the servant was sent to Miletos where his head was shaved where the message was retrieved.

Roman times

The Astragalus was the predecessor of today's cubical dice (Game piece's). A set of drilled Astragali could be used to conceal a message by means of weaving a thread through the holes in the pre-determined manner. These objects would be passed unnoticed as a toy. These were sometimes referred to as semagram.

Linguistic Steganography

The first linguistic Steganography methods appeared alongside Caesar's cipher. Null ciphers are form of open codes, where the plaintext is mixed with non-cipher material. The simplest to embed a steganogram in the first letters of words of the cover text.

Null Cipher Message sent from the us in WW1

President's Embargo ruling should have immediate notice. Grave situation affecting international law. Statement foreshadows ruin of many neutrals. Yellow journals unifying national excitement immensely.

Message: **PERSHING SAILS FROM NY JUNE 1**

The first publication on Steganography was entitled Steganographia written by Johannes Trithemius in the 16th century.

Girolamo Cardano proposed in the 15th century that a method of hiding secret messages by using a grid. A Cardan grille was a sheet of parchment with apertures for writing text. The text in the holes formed the steganogram which was composed into innocent looking text

World War 1

Sympathetic inks were first written of this method is attributed to Ovid who postulated using milk and soot as the corresponding developer. During World War 1 common pairs included phenolphthalein and ammonia or sodium carbonate.

In World War 2 the Germans spelled out a secret message by means of pricking pinholes above and below the letters in a newspaper article. The Nazi's used this technique as late as World War 2 dotting appropriate letters with sympathetic ink.

Spread Spectrum technologies

Spread Spectrum technologies is presently considered as means resilience to interference and noise. It originates back in 1941 when Hedy Lamarr and George Antheil proposed a Frequency Hopping radio system for torpedos.

Jargon Code

Velvalee Dickinson also known as The Doll Woman owned a doll shop in New York, who informed the Japanese troops about the location of American vessels. She used Jargon code and wrote letters about dolls, which served as cover meaning for warships.

Microdots

During the cold war innocent letters were embedded so called microdots. Microdots are miniature images or photographs developed in microscope size. They passed unnoticed as punctuation marks.

Subliminal Channels

Subliminal channels capacitate Steganography communication over an insecure channel with the aid of digital signatures. This is usually achieved by presenting the values of random parameters.

Subliminal channels base on Steganographic exploitation of a cryptographic protocol.

In 1984 Gustavus Simmons formulated the Prisoner's Problem. Two accomplices are arrested in separate cells. They can communicate via a warden who can consider the contents of their communication. The prisoners are to agree on an escape plan without raising suspicion of the warden. The solution is to create a subliminal channel.

Steganography in written text

The content can be embedded in print matter or in text. In 2004 it was revealed that several printers manufacture use steganography to hide information about printer serial numbers and the manufacturing code. This information may be used to track counterfeits.

Type spacing and offsetting

This embedding can also be performed by means of altering the appearance of text.

This is achieved by:

- Skewing
- Altering spacing
- Offsetting
- Font colour alterations

Digital Steganography

Concealment of messages in:

- Audio files
- Video files
- Digital images
- Text files
- Web content

Network Steganography

This is a new trend in Steganography. The information hiding technique which can be utilised to exchange steganograms in communication networks.

There are two types of Network Steganography. Intra-protocol Steganography which is a method that utilise one protocol to conceal information and inter-protocol Steganography which is a method that utilise more than one protocol to conceal information.

(<http://stegano.net/tutorial/steg-history.html>)

How is Steganography on an Image

The basic function of Steganography coding:

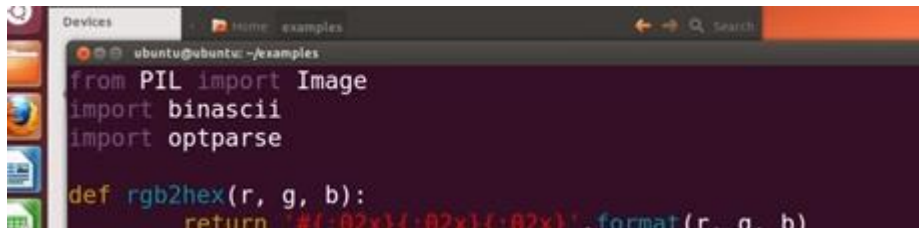
First open an image and look at its pixels in hexadecimal. If the pixel channel blue falls in the range of 0-5, the 1 bit of the information is stored. This ends the stream with delimiter of fifteen 1's and a 0 to take up to two bytes.

When it's time to retrieve it, it pulls all the blue bits of 0 and 1's until the stream obtains the delimiter of fifteen 1's and a 0.

A program is created to hide our message, using several functions for manipulating our data. An encode, decode, hide and retrieve.

The main will handle the arguments and whether to store or retrieve data.

Import a Library



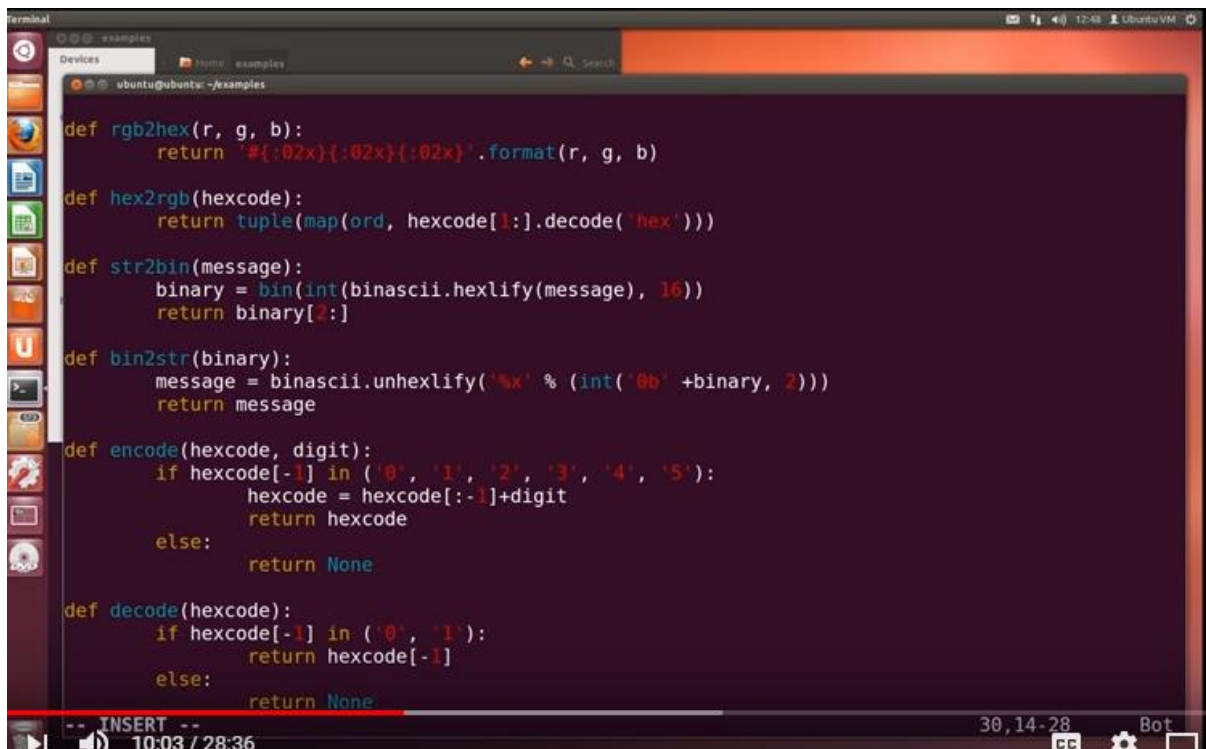
```
from PIL import Image
import binascii
import optparse

def rgb2hex(r, g, b):
    return '#{:02x}{:02x}{:02x}'.format(r, g, b)
```

First, libraries are imported. The PIL import Image is a Python Import Library to allow us to use images. The binascii is the binary library and the optparse is required.

(<https://www.youtube.com/watch?v=q3eOOMx5qoo>)

Explaining Decoding and Encoding Functions:



```
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

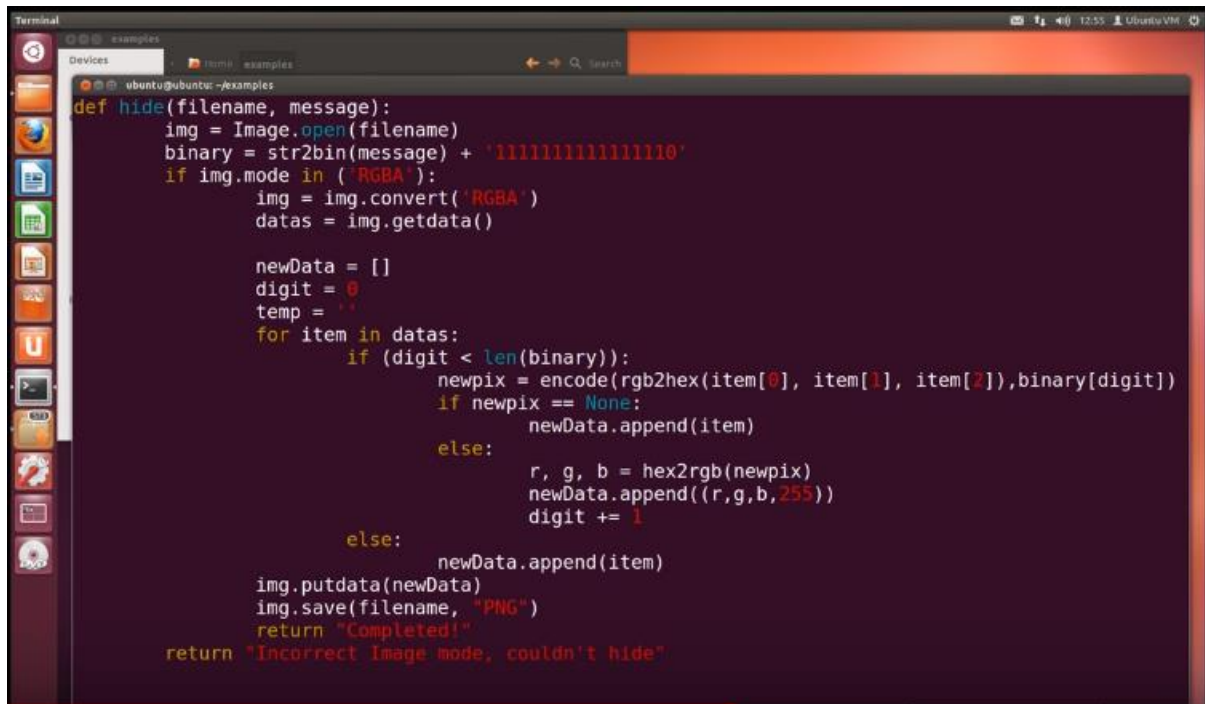
def encode(hexcode, digit):
    if hexcode[-1] in ('0', '1', '2', '3', '4', '5'):
        hexcode = hexcode[:-1] + digit
        return hexcode
    else:
        return None

def decode(hexcode):
    if hexcode[-1] in ('0', '1'):
        return hexcode[:-1]
    else:
        return None
```

A function is created to convert RGB to hex, and hex to RGB. The RGB is the colours of the image **red** **green** and **blue**. A function is created to convert string to binary and binary back to string. The encode function takes the first 0-5 **blue** and turns them to hexadecimal. The encode function is a reverse of the decode.

(<https://www.youtube.com/watch?v=q3eOOMx5qoo>)

Encoding:



```
def hide(filename, message):
    img = Image.open(filename)
    binary = str2bin(message) + '111111111111110'
    if img.mode in ('RGBA'):
        img = img.convert('RGB')
        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"
```

In the hide function we take a file and message. The file is the image used. Binary = string2bin(message) takes your fifteen 1's and 0. This creates a path to store binary inside the image. The fifteen 1's and a 0 is our delimiter.

The if statement is to find out if the image is editable. If so it will convert the RGB to and retrieve the data.

The for item in data will transform the pixels with the data. The 2 else is if the picture is not editable or couldn't hide the message.

Our Returns will display whether or not the objective was completed.

(<https://www.youtube.com/watch?v=q3eOOMx5qoo>)

Retrieve Data Function:

```
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"
```

The retrieve function is a reverse of the hide. It takes the RGB pixels and retrieves the data from it.

If it is successful it will return the data that was hidden. If not, it returns Incorrect image mode couldn't retrieve.

(<https://www.youtube.com/watch?v=q3eOOMx5qoo>)

Main Method:

```
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()
```

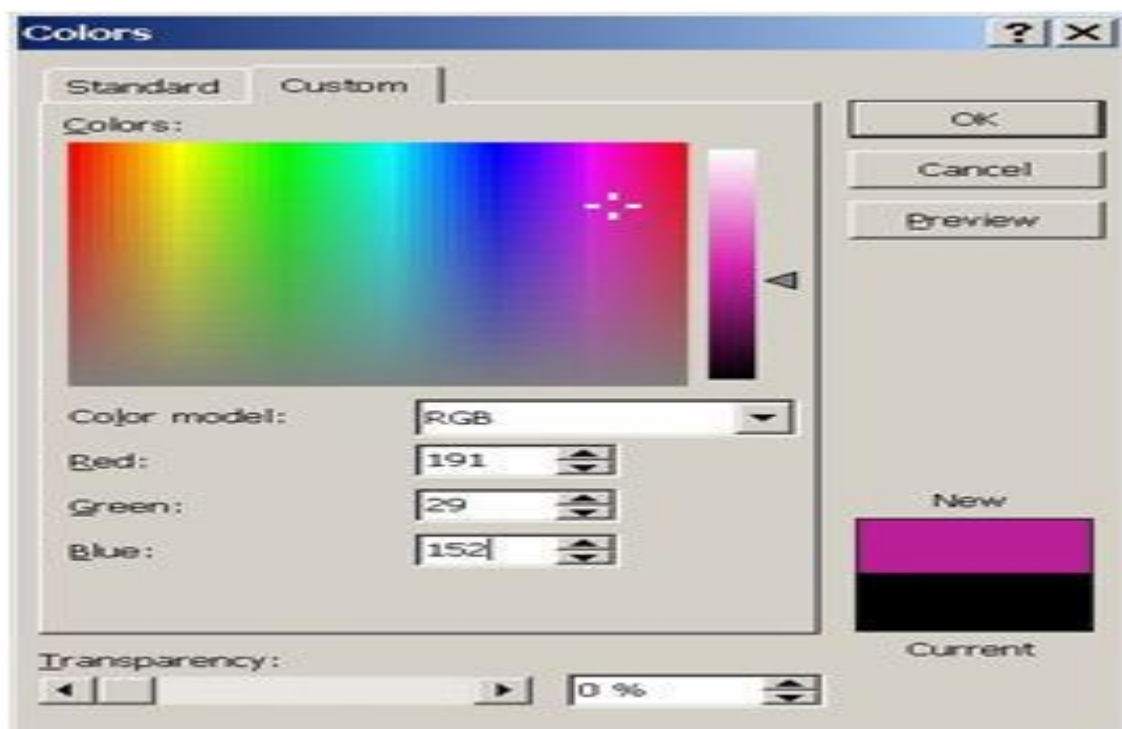
The main method the main method lets us choice which function we run. -e encode or -d decode.

It gives -e the function to hide a string a -d to retrieve the string. The if else statement take text and hides it if entered or else it will retrieve it.

The main functionality of the program in the encoding is to take a string turn it to binary, turn the binary to a hexadecimal and the retrieve function decodes it and turns it from hex to binary and the binary back to a string.

(The coding was done in python on ubuntu software)

(<https://www.youtube.com/watch?v=q3eOOMx5qoo>)



The figure above shows the red green and blue (RGB) level selected colour. RGB components is specified by a single byte. Each colour values intensity varies from 0-255. Shades denoted by red level of 191 (hex BF), green level of 29 (hex 1D) and blue level of 152 (hex 98)

Most digital images today support 24-bit. Each picture element (pixel) is encoded in 24 bits, comprised of RGB bytes. There are other 8 bits/pixels. Each 8 bits encoded where the value points to a 24-bit colour entry. The method limits the unique number of colours in each image to 256(2 to the power of 8).

(Johnson and Jajodia 1998A).

The transformation of Steganography

The techniques of Steganography have become more sublime across the ages, but the main principle remained the same. Every secret message is carried within some other entity and the communication can remain concealed.

Tools for Steganography

There are many tools available online for Steganography. These software tools let a user encode decode for detecting Steganography. There are tools pacific for different Steganography, like OpenStego for image files that cover bitmap image files (BMP) and Portable Network Graphics (PNG). DeepSound for Audio that cover Audio CD, APE, MP3. OpenPuff for video files. DarkCryptC for document files, and other sources such as EXE, DILL and NTFS.

Conclusion

Steganography has been around for hundreds of years and developed in many ways, but it still has the same function. For communication between to parties were a third party is unaware of its existence.

Steganography is used day to day. There have been reports that terrorist organization behind the September 11 attacks in New York City, Washington D.C, and Pittsburgh used steganography as one of their means of communication. Two books are written on the attacks. Attacks and Countermeasures by (N.F.Johnson, Z.Duric and S.Jajodia) and Information Hiding Techniques for Steganography and Digital Watermarking's by (S.Katzenbeisser and F.A.P Petitcolas).

References

Gary C. Kessler

September 2001 (Steganography: Hiding Data within Data)

Security and Cryptography

Affiliated Faculty: James Aspnes, Joan Feigenbaum, Mike Fischer, Zhong Shao.

University of Yale

Gary C. Kessler

February 2004 (updated February 2015)

Johnson, N. F. and Jajodia, S. Exploring steganography: Seeing the unseen, *Computer* (1998A) 31(2):26-34.

Johnson, N. F. and Jajodia, S. Steganalysis of images created using current steganography software.

Bauer, F. L. *Decrypted Secrets: Methods and Maxims of Cryptology*, 3rd ed. Springer-Verlag, New York, 2002.

Weisstein, Eric W. "Least Significant Bit."

Information Hiding Techniques for Steganography and Digital Watermarking, edited by S.

Katzenbeisser and F.A.P. Petitcolas (Artech House Books, 2000)

Attacks and Countermeasures by N.F. Johnson, Z. Duric, and S. Jajodia (Kluwer Academic Publishers, 2000)

Websites visited

<https://www.garykessler.net/library/steganography.html>

<https://www.itworld.com/article/2826840/crash-course-digital-steganography.html>

https://www.garykessler.net/library/fsc_stego.html

<http://www.jjtc.com/Steganography/>

<http://www.jjtc.com/ihws98/jjgmu.html>.

<http://www.jjtc.com/pub/r2026.pdf>.

<https://manytools.org/hacker-tools/steganography-encode-text-into-image/>

<http://resources.infosecinstitute.com/steganography-and-tools-to-perform-steganography/#gref>

http://embeddedsd.net/OpenPuff_Steganography_Home.html/

Watched videos on Steganography

https://www.youtube.com/watch?v=z_ypj5q5fzE – Principles of Steganography

<https://www.youtube.com/watch?v=osNWSGsFOvA> -

Network Steganography and Anomaly Detection

<https://www.youtube.com/watch?v=QoaGdmdjccU> - Steganography

https://www.youtube.com/watch?v=SWnPKI3_7h8 – What is Steganography? What does it mean & explanation

<https://www.youtube.com/watch?v=q3eOOMx5qoo> – Steganography Tutorial – Hiding text inside Images

<https://www.youtube.com/watch?v=c8LrqAm3CfA> – Steganography History

https://www.youtube.com/watch?v=d3_2m8ICxOw -History of Steganography

<https://www.youtube.com/watch?v=VuZueOE-fDs> – Cyber Security - Risk of Steganography [Fundamental Concept]

<https://www.youtube.com/watch?v=P1suwm9zYel> – Steganography: Hiding an image in Audio

<https://www.youtube.com/watch?v=teShYhts2So> – How to hide Secret Messages in Audio | Audio Steganography

<https://www.youtube.com/watch?v=TWEXCYQKyDc> –

[Shttps://www.youtube.com/watch?v=z_ypj5q5fzE](https://www.youtube.com/watch?v=z_ypj5q5fzE)crets of Hidden in images (Steganography) - Computerphile

