IMAGE STEGANOGRAPHY

A Project Report
Submitted in partial fulfillment of the requirement for the award of
Degree of Bachelor of Technology in Electronics & Communication

Submitted to

Rajiv Gandhi Proudyogiki Vishwavidhyalaya, Bhopal (M.P.)

Minor Project Report
Submitted by

SUDHANSHU SUNNY (0157EC191114)   SHUBHAM KUMAR (0157EC191110)

SHIVANSH PANDEY(0157EC191108)

Under the supervision of
Prof. Rohit Kumar Rathor   **Prof. Rahul Sharma**

Lakshmi Narain College of Technology & Science, Bhopal

Session 2020-2021

Department of Electronics & Communication Engineering

Lakshmi Narain College of Technology & Science, Bhopal

Department of Electronics & Communication Engineering

CERTIFICATE

This is to certify that the work embodied in this project entitled "**Image Steganography**" has been satisfactorily completed **by Sudhanshu Sunny, Shubham Kumar, Shivansh Pandey**. It is a bona fide piece of work, carried out under our/my guidance in the Department of **Electronics & Communication Engineering, Lakshmi Narain College Of Technology & Science**, Bhopal for the partial fulfillment of the **Bachelor of Technology** during the academic year 2021-2022.

Under the supervision of

Prof. Rohit Kumar Rathor	**Prof. Rahul Sharma**
(Project Guide)	(Project In charge)

Approved By

____

Dr. Soheb Munir

(Professor & Head of Department)

Department of electronics and communication engineering

Lakshmi Narain College of Technology & Science, Bhopal

Department of Electronics & Communication Engineering

CERTIFICATE OF APPROVAL

This foregoing project work is hereby approved as a creditable study of an Engineering subject carried out and presented in a manner satisfactorily to warranty its acceptance as a prerequisite to the degree for which it has been submitted. It is understood that by this approval the undersigned do not necessarily endorse or approve any statement made, opinion expressed or conclusion drawn therein, but approve the project only for the purpose for which it has been submitted.

Approved by

…………………………

Dr. Soheb Munir

(Head of the department)

Lakshmi Narain College of Technology & Science, Bhopal

Department of Electronics & Communication Engineering

DECLARATION

We **Sudhanshu Sunny, Shubham Kumar, Shivansh Pandey** students of Bachelor of Technology, Branch Electronics & Communication Engineering, **Lakshmi Narain College Of Technology & Science, Bhopal** hereby declare that the work presented in this Minor Project is outcome of our own work, is bonafide, correct to the best of our knowledge and this work has been carried out taking care of engineering ethics. The work presented does not infringe any patented work and has not been submitted to any university for the award of any degree or any professional diploma.

Sudhanshu Sunny (0157EC191114) ___________________

Shubham Kumar (0157EC191110) ___________________

Shivansh Pandey (0157EC191108) ___________________

Date : __________

* * * * * * *

Lakshmi Narain College of Technology and Science, Bhopal

Department of Electronics & Communication Engineering

ACKNOWLEDGEMENT

Reasons can be given & they can be many but nothing can replace the efforts of numerous people behind the work, put in by the creator, giving us constant support all the way.

We are greatly indebted to **Dr. Soheb Munir, H.O.D of Electronics and Communication Engineering, LNCTS, Bhopal** who has been there as our supervisor for the whole years of engineering and has been unstinting in his support and constructive critique throughout the progress of the dissertation. Also, we owe our profound gratitude to **Dr. Ashok Kumar Rai, Director of Administrator, LNCT group, Bhopal** and to Dr **Amitbodh Upadhyaya, OSD, LNCTS** for providing us with all the necessary facilities and to **Dr. Amit Sachan, Principal, LNCTS.**

We would like to express our deep sense of gratitude to Prof. Rahul Sharma Project In-charge of Electronics and Communication Engineering, for providing us all support and guidance all along, till the completion of our project work by providing all the necessary information for developing a good system. Many thanks are also to **Prof. Rohit Kumar Rathor, Project Guide** who started with us down this road and whose encouragement, suggestion and very constructive criticism have contributed immensely to the evolution of our ideas on project.

We thank our parents, family, and friends for bearing with us throughout the course of this project and for the opportunity they provided in undergoing this course in such a **prestigious institution.**

Sudhanshu Sunny	**(0157EC191114)**
Shubham Kumar	**(0157EC191110)**
Shivansh Pandey	**(0157EC191108)**

CONTENTS

List of Figures

About Us

	My name is Sudhanshu Sunny. At present I am pursuing my Bachelor of Technology in Electronics and Communication in Lakshmi Narain College of Technology and Science, Bhopal. My current CGPA is 9.40 . I have done my schooling from Nitesh Kumar Samarak H/S, Paharpur Soi, Vaishali Bihar (in 12th) and Indian Public School, Hajipur, Vaishali, Bihar(in 10th). I am an active learner and hard worker.
	My name is Shivansh Pandey, presently I am pursuing my Bachelor of technology in Electronics and Communication from Lakshmi Narain College of Technology and Science, Bhopal. Currently my C.G.P.A is 8.9 .I did my schooling from G.V.N the global school (12th) and DPS, Bhopal(10th). I am a active learner and hard worker.
	My name is Shubham Kumar, presently I am pursuing my Bachelor of technology in Electronics and Communication from Lakshmi Narain College of Technology and Science, Bhopal. Currently my C.G.P.A is 8.5 .I did my schooling from Foundation School, Buxar, Bihar (12th) and St Paul's School, Sasaram, Bihar (10th). I am a active learner and hard worker.

Abstract

Image Steganography is the process of hiding information which can be text, image or video inside a cover image. The secret information is hidden in a way that it not visible to the human eyes. Deep learning technology, which has emerged as a powerful tool in various applications including image steganography, has received increased attention recently. The main goal of this paper is to explore and discuss various deep learning methods available in image steganography field. Deep learning techniques used for image steganography can be broadly divided into three categories - traditional methods, Convolutional Neural Network-based and General Adversarial Network-based methods. Along with the methodology, an elaborate summary on the datasets used, experimental set-ups considered and the evaluation metrics commonly used are described in this paper. A table summarizing all the details are also provided for easy reference. This paper aims to help the fellow researchers by compiling the current trends, challenges and some future direction in this field.

CHAPTER 1

Introduction

1.1 Introduction

The word Steganography is derived from two Greek words- ‘stegos’ meaning ‘to cover’ and ‘grayfia’, meaning ‘writing’, thus translating to ‘covered writing’, or ‘hidden writing’. Steganography is a method of hiding secret data, by embedding it into an audio, video, image, or text file. It is one of the methods employed to protect secret or sensitive data from malicious attacks.

The first recorded use of the term was in 1499 by Johannes Trithemius in his Steganographia, a treatise on cryptography and steganography, disguised as a book on magic. Generally, the hidden messages appear to be (or to be part of) something else: images, articles, shopping lists, or some other cover text. For example, the hidden message may be in invisible ink between the visible lines of a private letter. Some implementations of steganography that lack a shared secret are forms of security through obscurity, and key-dependent steganographic schemes adhere to Kerckhoffs's principle

How is it different from cryptography?

Cryptography and steganography are both methods used to hide or protect secret data. However, they differ in the respect that cryptography makes the data unreadable, or hides the meaning of the data, while steganography hides the existence of the data.

In layman’s terms, cryptography is similar to writing a letter in a secret language: people can read it, but won’t understand what it means. However, the existence of a (probably secret) message would be obvious to anyone who sees the letter, and if someone either knows or figures out your secret language, then your message can easily be read.

If you were to use steganography in the same situation, you would hide the letter inside a pair of socks that you would be gifting the intended recipient of the letter. To those who don’t know about the message, it would look like there was nothing more to your gift than the socks. But the intended recipient knows what to look for, and finds the message hidden in them.

Similarly, if two users exchanged media files over the internet, it would be more difficult to determine whether these files contain hidden messages than if they were communicating using cryptography.

Cryptography is often used to supplement the security offered by steganography. Cryptography algorithms are used to encrypt secret data before embedding it into cover files.

Image Steganography –

As the name suggests, Image Steganography refers to the process of hiding data within an image file. The image selected for this purpose is called the cover image and the image obtained after steganography is called the stego image.

An image is represented as an N*M (in case of greyscale images) or N*M*3 (in case of color images) matrix in memory, with each entry representing the intensity value of a pixel. In image steganography, a message is embedded into an image by altering the values of some pixels, which are chosen by an encryption algorithm. The recipient of the image must be aware of the same algorithm in order to know which pixels he or she must select to extract the message.

Detection of the message within the cover image is done by the process of steganalysis. This can be done through comparison with the cover image, histogram plotting, or noise detection. While efforts are being invested in developing new algorithms with a greater degree of immunity against such attacks, efforts are also being devoted towards improving existing algorithms for steganalysis, to detect the exchange of secret information between terrorists or criminal elements.

1.2 Block Diagram

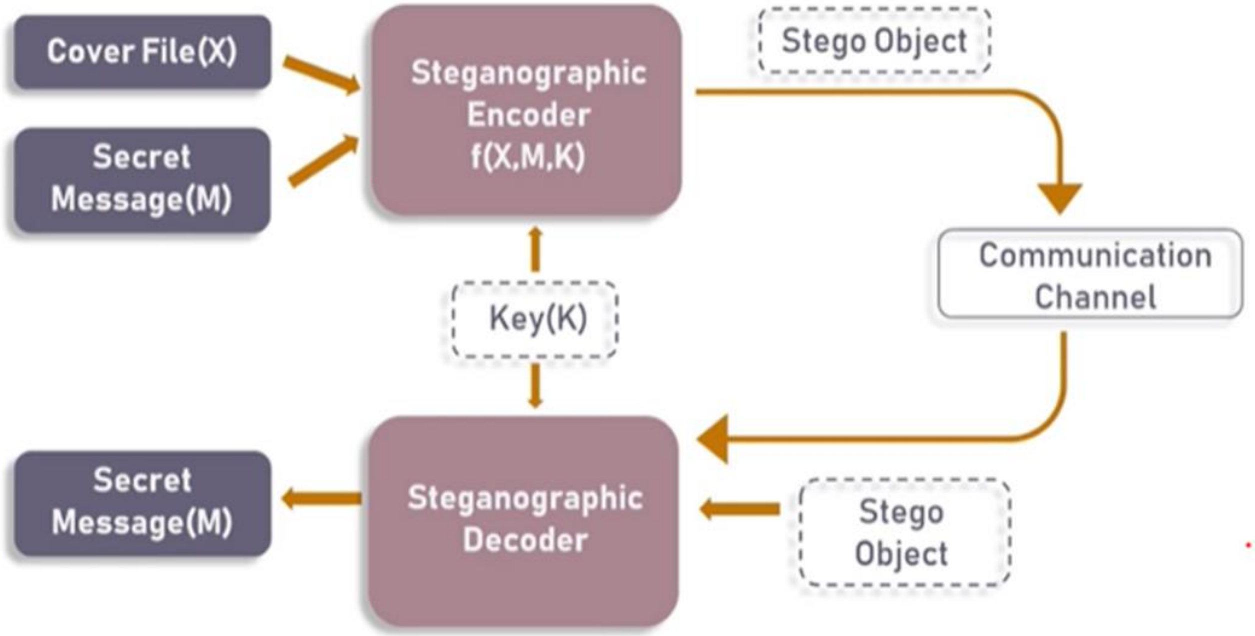


Figure 1: Block Diagram of the Project

As seen in the above image, both the original image file(X) and secret message (M) that needs to be hidden are fed into a steganographic encoder as input. Steganographic Encoder function, f(X,M,K) embeds the secret message into a cover image file by using techniques like least significant bit encoding. The resulting stego image looks very similar to your cover image file, with no visible changes. This completes encoding. To retrieve the secret message, stego object is fed into Steganographic Decoder.

CHAPTER 2

Principle

2.1 Image Steganography

Hiding information inside images is a popular technique nowadays. An image with a secret message inside can easily be spread over the World Wide Web or in newsgroups. The use of steganography in newsgroups has been researched by German steganographic expert Niels Provos, who created a scanning cluster which detects the presence of hidden messages inside images that were posted on the net. However, after checking one million images, no hidden messages were found, so the practical use of steganography still seems to be limited.

To hide a message inside an image without changing its visible properties, the cover source can be altered in ”noisy” areas with many color variations, so less attention will be drawn to the modifications. The most common methods to make these alterations involve the usage of the least-significant bit or LSB, masking, filtering and transformations on the cover image. These techniques can be used with varying degrees of success on different types of image files.

2.2 Least Significant Bit Steganography

We can describe a **digital image** as a finite set of digital values, called pixels. Pixels are the smallest individual element of an image, holding values that represent the brightness of a given color at any specific point. So we can think of an image as a matrix (or a two-dimensional array) of pixels which contains a fixed number of rows and columns.

Least Significant Bit (LSB) is a technique in which the last bit of each pixel is modified and replaced with the secret message’s data bit.

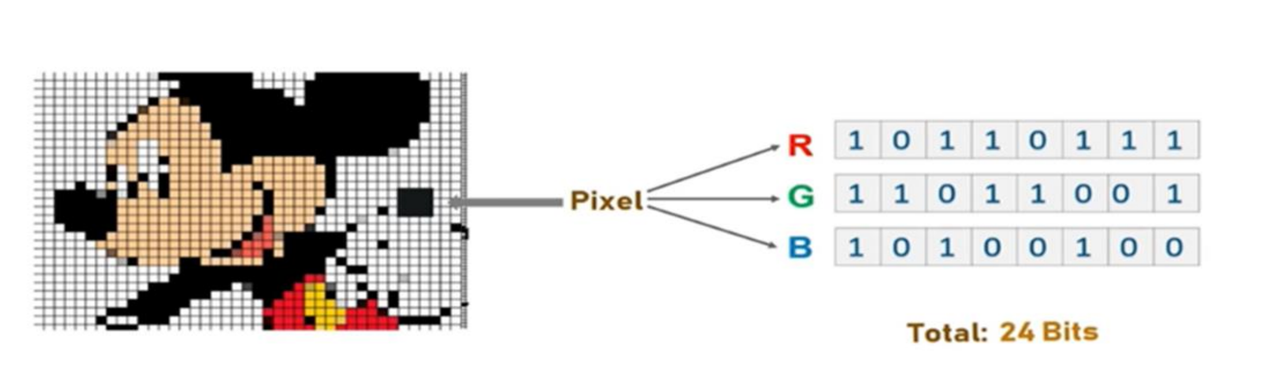


Figure 2: LSB Example

From the above image it is clear that, if we change MSB it will have a larger impact on the final value but if we change the LSB the impact on the final value is minimal, thus we use least significant bit steganography.

CHAPTER 3

Working

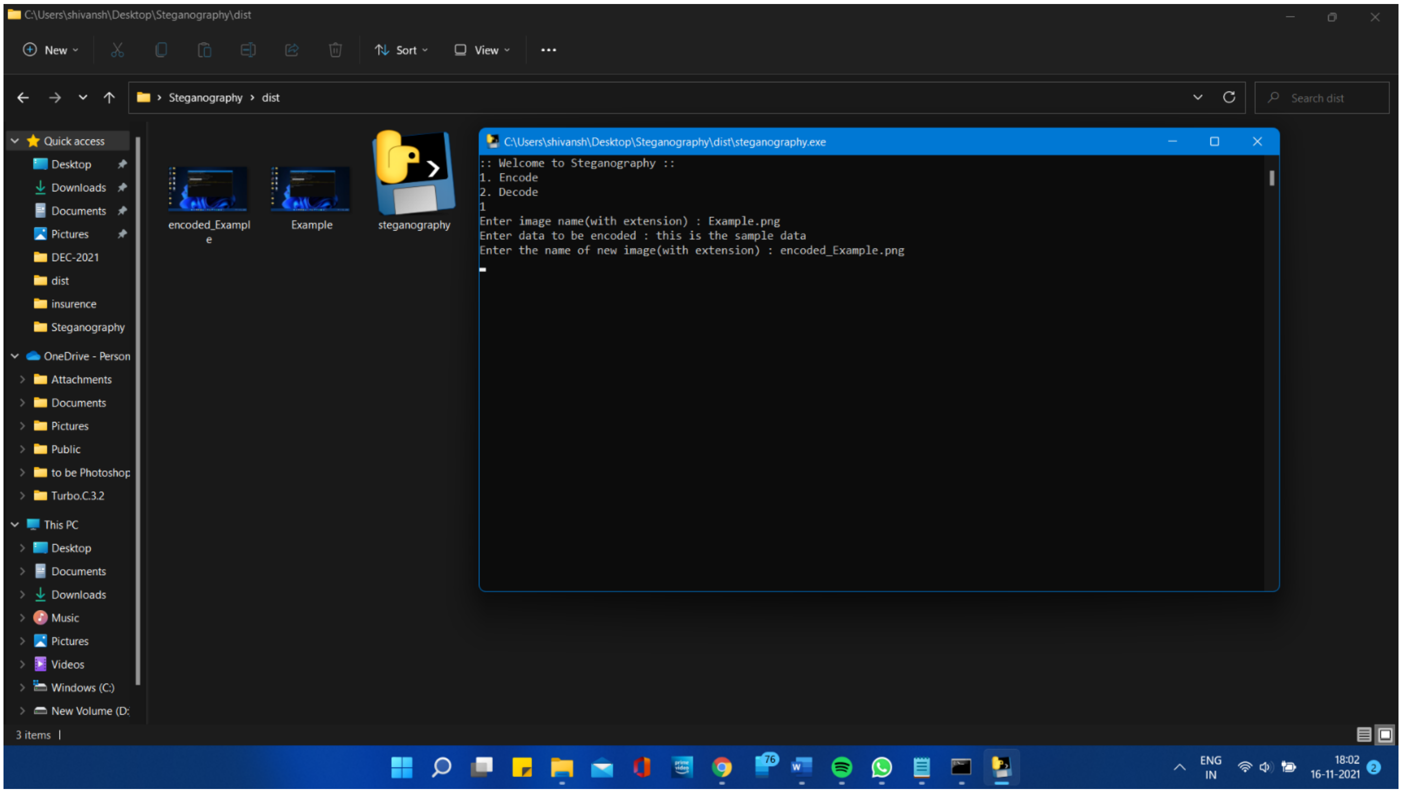
3.1 Encode the data :

Every byte of data is converted to its 8-bit binary code using ASCII values. Now pixels are read from left to right in a group of 3 containing a total of 9 values. The first 8-values are used to store binary data. The value is made odd if 1 occurs and even if 0 occurs.
For example :
Suppose the message to be hidden is ‘ Hii ‘. Since the message is of 3-bytes, therefore, pixels required to encode the data is 3 x 3 = 9. Consider a 4 x 3 image with a total 12-pixels, which are sufficient to encode the given data.

[(27, 64, 164), (248, 244, 194), (174, 246, 250), (149, 95, 232),
(188, 156, 169), (71, 167, 127), (132, 173, 97), (113, 69, 206),
(255, 29, 213), (53, 153, 220), (246, 225, 229), (142, 82, 175)]

ASCII value of ‘ H ‘ is 72 whose binary equivalent is 01001000.
Taking first 3-pixels (27, 64, 164), (248, 244, 194), (174, 246, 250) to encode. Now change the pixel to odd for 1 and even for 0. So, the modifies pixels are (26, 63, 164), (248, 243, 194), (174, 246, 250). Since we have to encode more data, therefore, the last value should be even. Similarly, ‘i‘ can be encoded in this image.
The new image will look like :

[(26, 63, 164), (248, 243, 194), (174, 246, 250), (148, 95, 231),
(188, 155, 168), (70, 167, 126), (132, 173, 97), (112, 69, 206),
(254, 29, 213), (53, 153, 220), (246, 225, 229), (142, 82, 175)]



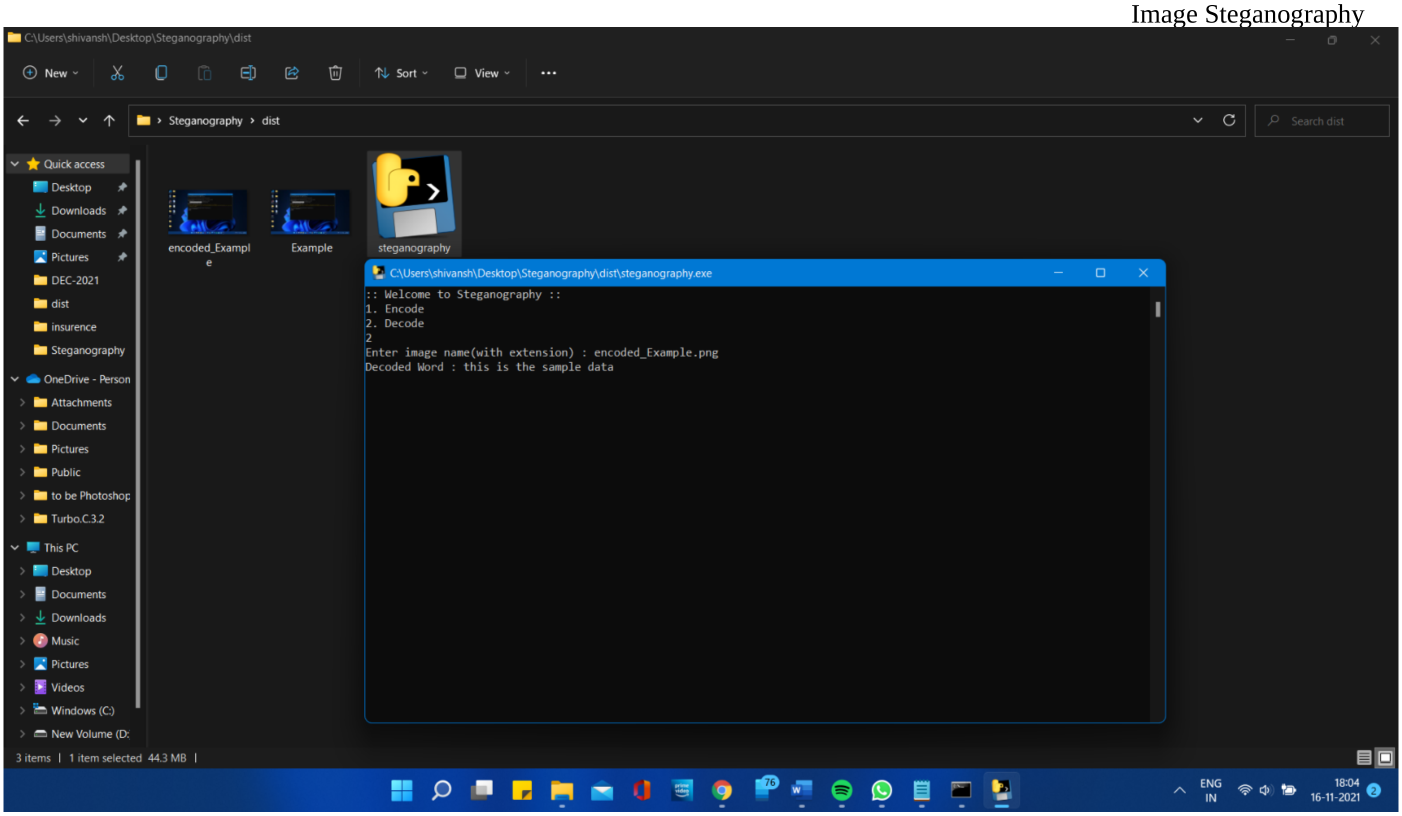
C:\Users\shivansh\Desktop\Steganography\dist\steganography.exe

```
:: Welcome to Steganography ::
1. Encode
2. Decode
1
Enter image name(with extension) : Example.png
Enter data to be encoded : this is the sample data
Enter the name of new image(with extension) : encoded_Example.png
```

Figure 3.1 : Example of how to encode data in image

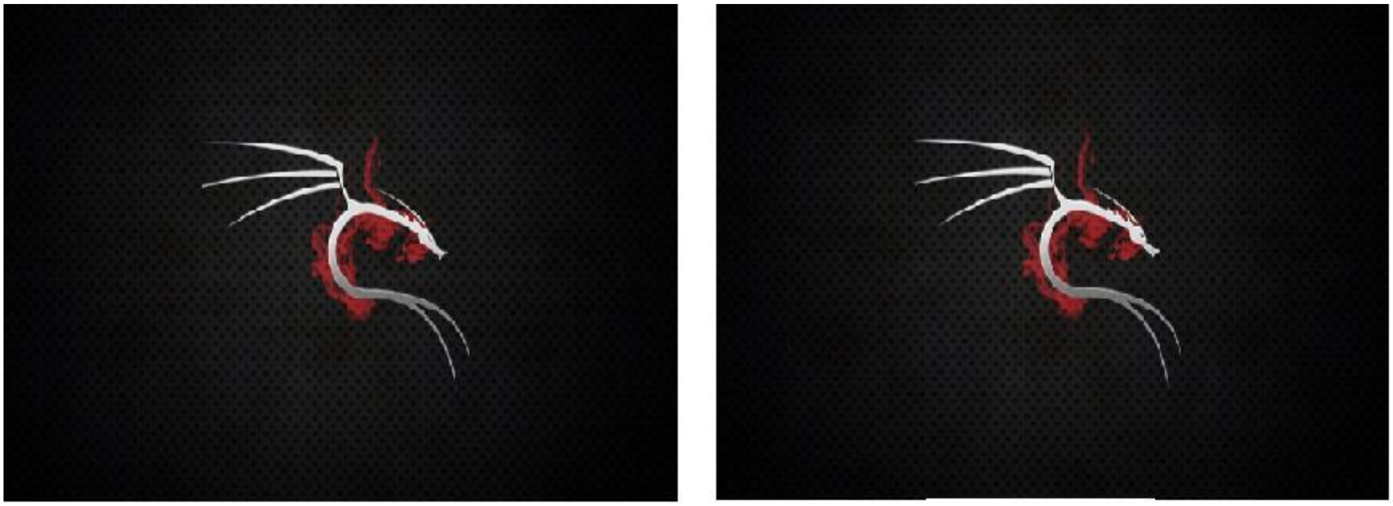
3.2 Decoding the data :

To decode, three pixels are read at a time, till the last value is odd, which means the message is over. Every 3-pixels contain a binary data, which can be extracted by the same encoding logic, where the 3-pixels are taken in the groups of 3, which brings the total to 9-bits at a time. In these 9-bits the first 8 are the ASCII code for a letter or a character in the ASCII table and the last one is ignored as that is the indicator whether to read further or not. The 8-bit segments are put in a list and converted to the corresponding letter or character which gives you the original message.



```
:: Welcome to Steganography ::
1. Encode
2. Decode
2
Enter image name(with extension) : encoded_Example.png
Decoded Word : this is the sample data
```

Figure 3.2 : Example of how to decode the information



Input Image

Output Image

Figure 3.3 : Initial and Final Images

CHAPTER-4

Source Code

```
def genData(data):

        # list of binary codes
        # of given data
        newd = []

        for i in data:
            newd.append(format(ord(i), '08b'))
        return newd

# Pixels are modified according to the
# 8-bit binary data and finally returned
def modPix(pix, data):

    datalist = genData(data)
    lendata = len(datalist)
    imdata = iter(pix)

    for i in range(lendata):

        # Extracting 3 pixels at a time
        pix = [value for value in imdata.__next__()[:3] +
                                imdata.__next__()[:3] +
                        imdata.__next__()[:3]]

for j in range(0, 8):
    if (datalist[i][j] == '0' and pix[j]% 2 != 0):
        pix[j] -= 1

    elif (datalist[i][j] == '1' and pix[j] % 2 == 0):
        if(pix[j] != 0):
            pix[j] -= 1
        else:
            pix[j] += 1
        # pix[j] -= 1

# Eighth pixel of every set tells
# whether to stop ot read further.
# 0 means keep reading; 1 means thec
# message is over.
if (i == lendata - 1):
    if (pix[-1] % 2 == 0):
        if(pix[-1] != 0):
            pix[-1] -= 1
        else:
            pix[-1] += 1

else:
    if (pix[-1] % 2 != 0):
        pix[-1] -= 1
```

```
            pix = tuple(pix)
            yield pix[0:3]
            yield pix[3:6]
            yield pix[6:9]

def encode_enc(newimg, data):
    w = newimg.size[0]
    (x, y) = (0, 0)

    for pixel in modPix(newimg.getdata(), data):

        # Putting modified pixels in the new image
        newimg.putpixel((x, y), pixel)
        if (x == w - 1):
            x = 0
            y += 1
        else:
            x += 1

# Encode data into image
def encode():
    img = input("Enter image name(with extension) : ")
    image = Image.open(img, 'r')

    data = input("Enter data to be encoded : ")
    if (len(data) == 0):
        raise ValueError('Data is empty')
    newimg = image.copy()
    encode_enc(newimg, data)

    new_img_name = input("Enter the name of new image(with extension) : ")
    newimg.save(new_img_name, str(new_img_name.split(".")[1].upper()))

# Decode the data in the image
def decode():
    img = input("Enter image name(with extension) : ")
    image = Image.open(img, 'r')

    data = ''
    imgdata = iter(image.getdata())

    while (True):
        pixels = [value for value in imgdata.__next__()[:3] +
                                imgdata.__next__()[:3] +
                                imgdata.__next__()[:3]]

        # string of binary data
        binstr = ''

        for i in pixels[:8]:
            if (i % 2 == 0):
                binstr += '0'
            else:
                binstr += '1'
        data += chr(int(binstr, 2))
        if (pixels[-1] % 2 != 0):
            return data

# Main Function
def main():
    a = int(input(":: Welcome to Steganography ::\n"
                        "1. Encode\n2. Decode\n"))
    if (a == 1):
        encode()

    elif (a == 2):
        print("Decoded Word : " + decode())
    else:
        raise Exception("Enter correct input")

# Driver Code
if __name__ == '__main__' :

    # Calling main function
    main()
    input()
```

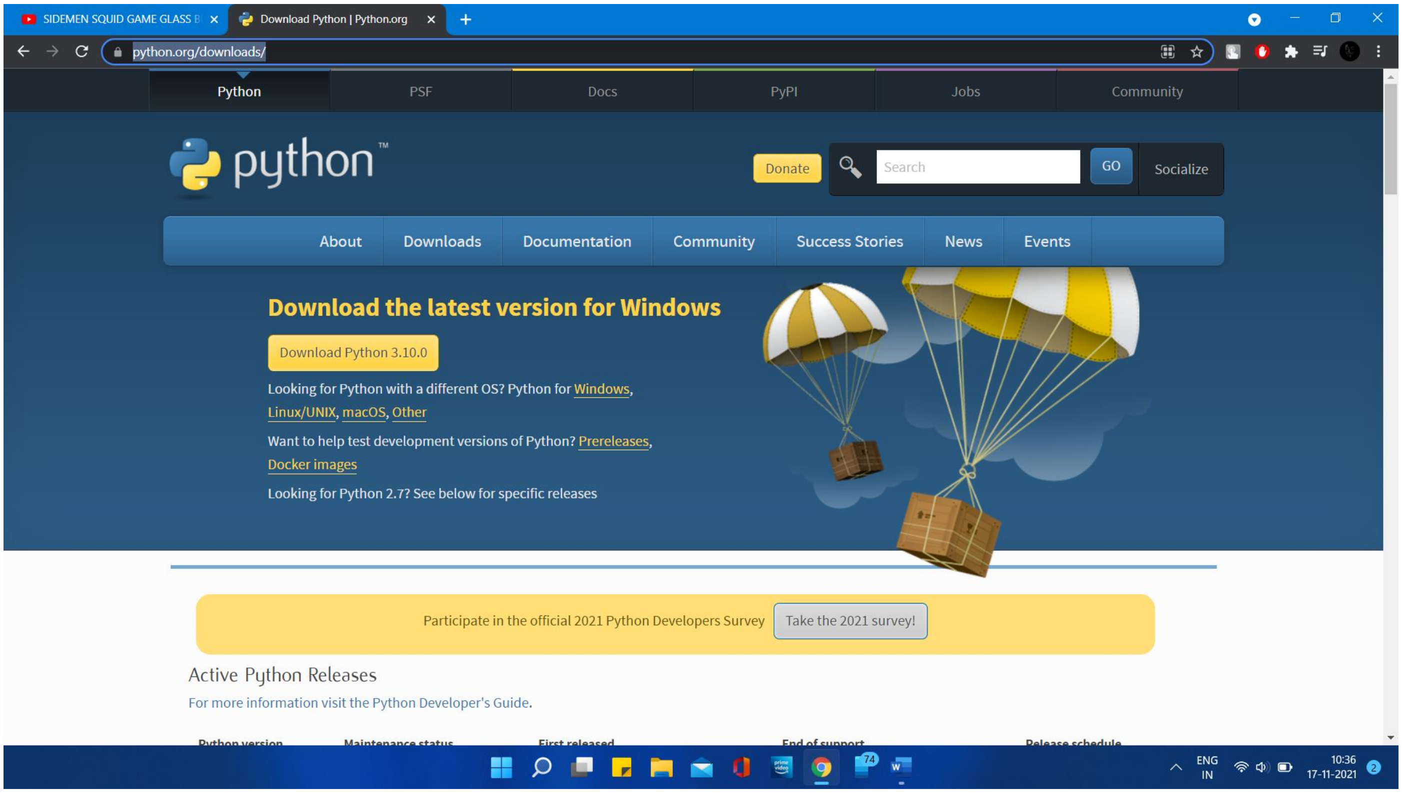
CHAPTER-5

Running the Code

5.1 Prerequisites of the Code :

Before you run the code you have make sure of a few things which are:

- You have to make sure that you have python 3 installed on your device , if you do not have python 3 installed you can download it from https://www.python.org/downloads/ and continue with the install wizard



Download the latest version for Windows

Looking for Python with a different OS? Python for Windows, Linux/UNIX, macOS, Other

Want to help test development versions of Python? Prereleases, Docker images

Looking for Python 2.7? See below for specific releases

Participate in the official 2021 Python Developers Survey

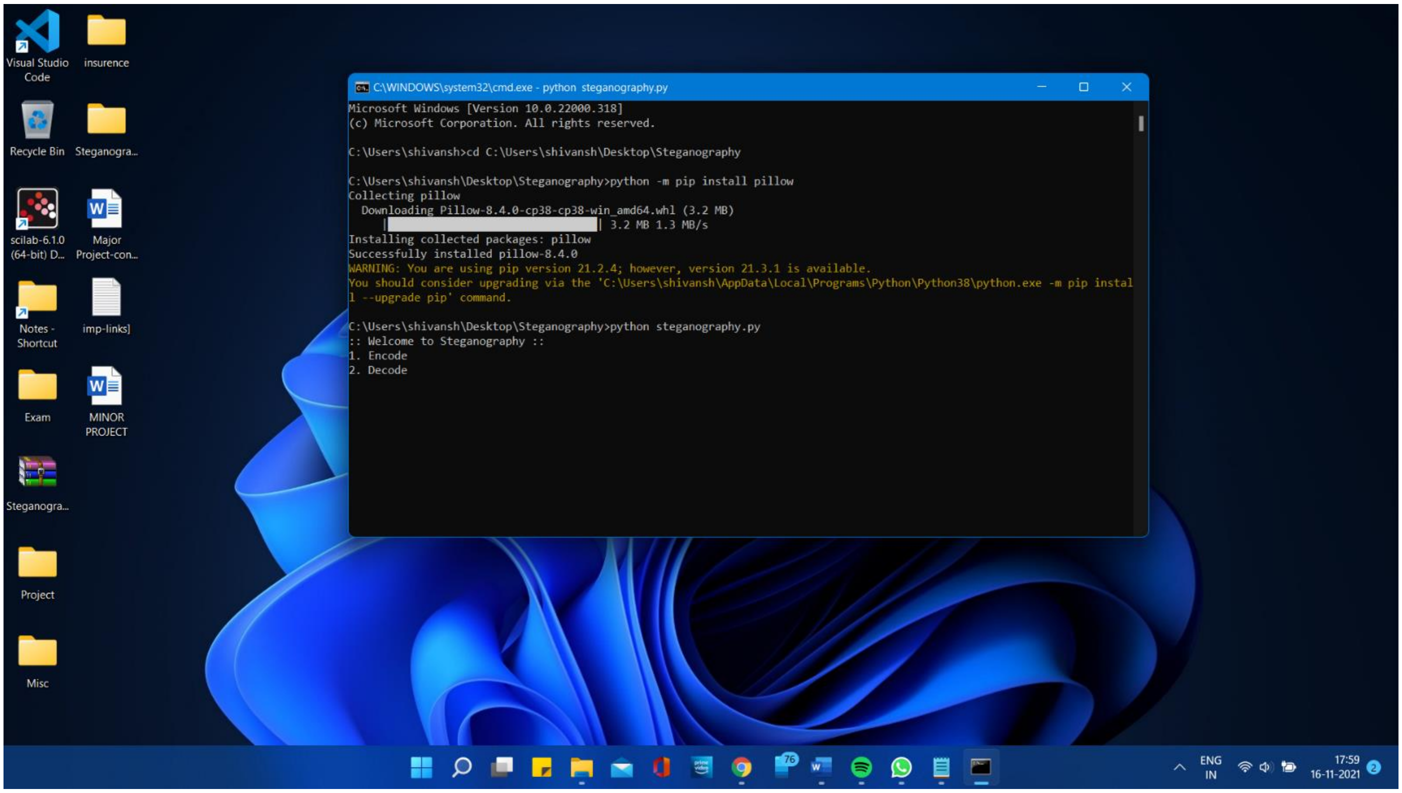
Active Python Releases

For more information visit the Python Developer's Guide.

Figure 5.1 : Python.org/downloads

- Next you have to install the pillow library for python, which can be done easily by command prompt
  You just have to use the pip module in python to download it which is

  python -m pip install pillow



```
Microsoft Windows [Version 10.0.22000.318]
(c) Microsoft Corporation. All rights reserved.

C:\Users\shivansh>cd C:\Users\shivansh\Desktop\Steganography

C:\Users\shivansh\Desktop\Steganography>python -m pip install pillow
Collecting pillow
  Downloading Pillow-8.4.0-cp38-cp38-win_amd64.whl (3.2 MB)
     |████████████████████████████████| 3.2 MB 1.3 MB/s
Installing collected packages: pillow
Successfully installed pillow-8.4.0
WARNING: You are using pip version 21.2.4; however, version 21.3.1 is available.
You should consider upgrading via the 'C:\Users\shivansh\AppData\Local\Programs\Python\Python38\python.exe -m pip instal
l --upgrade pip' command.

C:\Users\shivansh\Desktop\Steganography>python steganography.py
:: Welcome to Steganography ::
1. Encode
2. Decode
```

Figure 5.2 : Downloading Pillow

After which you can run the program easily by just clicking on it or by using command prompt , it is relatively easy after this and using these steps the program can be used anywhere at any time as it was intended to do from the beginning .

CHAPTER 6

Result and Conclusion

In this project We mainly concentrated on embedding the data into an image. We have designed the steganographic application which embedded the data into the image.

Normally, after embedding the data into the image, the image may lose its resolution. In the proposed approach, the image remains unchanged in its resolution as well in size.

The speed of embedding the data into the image is also high in the proposed approach such that the image is protected and the data to the destination is sent securely. For the decryption phase, I have used the same Python programming language for the purpose of designing.

There are many steganographic algorithms available like JSteg, F5 and LSB algorithms. I have used the Least Significant Bit algorithm in designing the steganographic application because LSB algorithm works efficiently when we consider bit map images .bmp files. The speed of embedding is also high when using LSB compared to the Steg algorithm.

CHAPTER 7

Shortcoming and Limitations

The main Shortcoming is the maximum size of the embedded data compared to the total data. If a piece of data is already very compressed it might be wholly impossible to embed additional data in it. And even under ideal conditions you will rarely get more than 20% out of the carrier data.

I assume of course that the data hidden is encrypted first, making it appear completely random even to statistical analysis programs. This reduces those (ideal) 30% by half on average.

So assume that you use a bunch of image files of moderate compression level as carrier medium. Lets say you get on average 15% out of it, lets say the total batch of images has a size of 1GB.

This means that after encrypting and embedding your data, you will be able to transport 75MB of hidden data. This is not a lot.

Steganography is in general only used in situations where there is no other alternative because the very fact that A and B are communicating would lead to grave consequences. Trying to swap and share files for example is not such a situation.

There are (at least) 2 limitations:

1) how much image/sound distortion is tolerable, before the user starts noticing something weird.

The whole purpose for the steganography is to make sure the message is hidden. So if the container media is too broken (noisy sound / picture) this might unveil the message.
But how do you define “too broken”? If you have an original, then it is a byte-by-byte comparison, and message is unmasked. If you don`t - then you have to rely on human senses. And here you have no objective criteria - someone can detect fuzz in a seemingly monotone part of the picture - and someone can`t.

You can use whatever technical means for determining the distortion rate, but for steganography you`d have to make a “blind” test on hidden message detection.

2) how much image/sound distortion is tolerable before the message is lost in the noises.

We rarely use unpacked WAV/FLAC files to share sound or lossless PNG/GIF for images. We use MP3s, JPEGs, MPEG2s and h264 when exchanging music, pictures and videos - and these are lossy. And unpacking/repacking an MP3 might lose some (or all) of the message bits.

CHAPTER 8

Future Scope

In the present world, the data transfers using internet is rapidly growing because it is so easier as well as faster to transfer the data to destination. So, many individuals and business people use to transfer business documents, important information using internet.

Security is an important issue while transferring the data using internet because any unauthorized individual can hack the data and make it useless or obtain information un-intended to him.

The future work on this project is to improve the compression ratio of the image to the text. This project can be extended to a level such that it can be used for the different types of image formats like .bmp, jpeg. .tif etc., in the future. The security using Least Significant Bit Algorithm is good but we can improve the level to a certain extent by varying the carriers as well as using different keys for encryption and decryption.

REFERENCES

- https://www.geeksforgeeks.org/text-extraction-from-image-using-lsb-based-steganography/?ref=lbp

- https://www.geeksforgeeks.org/image-steganography-in-cryptography/?ref=lbp

- https://www.researchgate.net/publication/308646775_An_introduction_to_steganography_methods

- https://en.wikipedia.org/wiki/Steganography

- https://pypi.org/project/Pillow/

ANNEXURE I

ENVIRONMENT AND SUSTAINABILITY

- This project reduces consumption of various types of energy.
- First is reduction of human resources used in process.
- Secondly reduction of paper and ink.
- It also saves the fund of agencies who sent messages to other organization.
- It is handle able by single or least group of person.
- Time taken for information to reach the destion is very less.
- It also reduces the need for transportation.
- Means it reduces pollution and is eco-friendly.

ANNEXURE II

Impact On Society

Application of Steganography

- Defense organization: security from enemies
- Intelligence Agencies: security of person's private information
- Government Agencies: store critical data like criminal record
- Medical: patient's details are embedded within image
- Smart identity cards: personal information is embedded into photo
- Secure Private Files and Documents.
- Hide Passwords and Encryption Keys.
- Transport Highly Private Documents between International Governments.
- Transmit message/data without revealing the existence of available message.

Advantages of Image Steganography:

- Difficult to detect. Only receiver can detect.
- Can be applied differently in digital image, audio and video file.

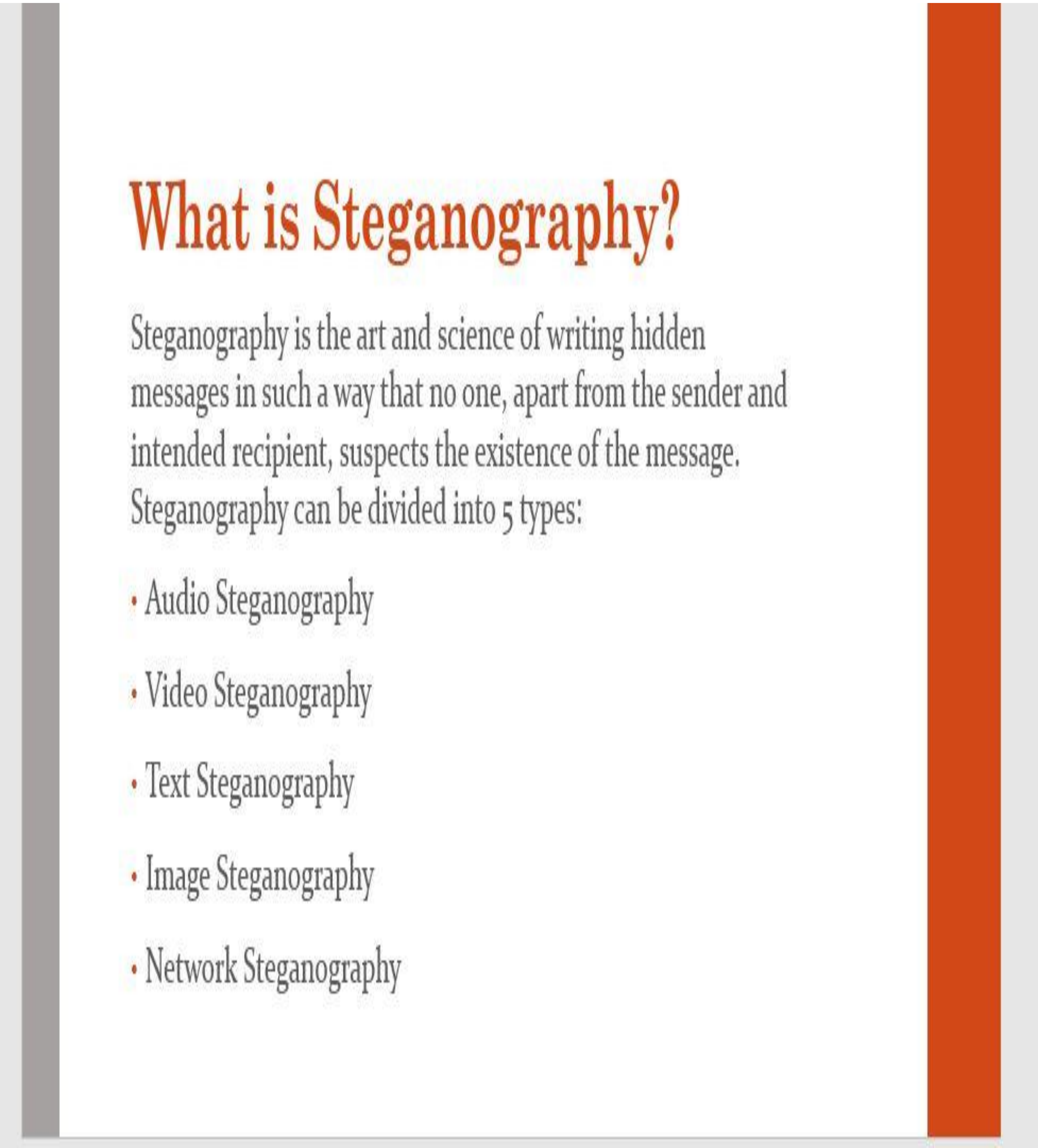
Cost of Project

Rs 100/-

ANNEXURE III

PRESENTATION

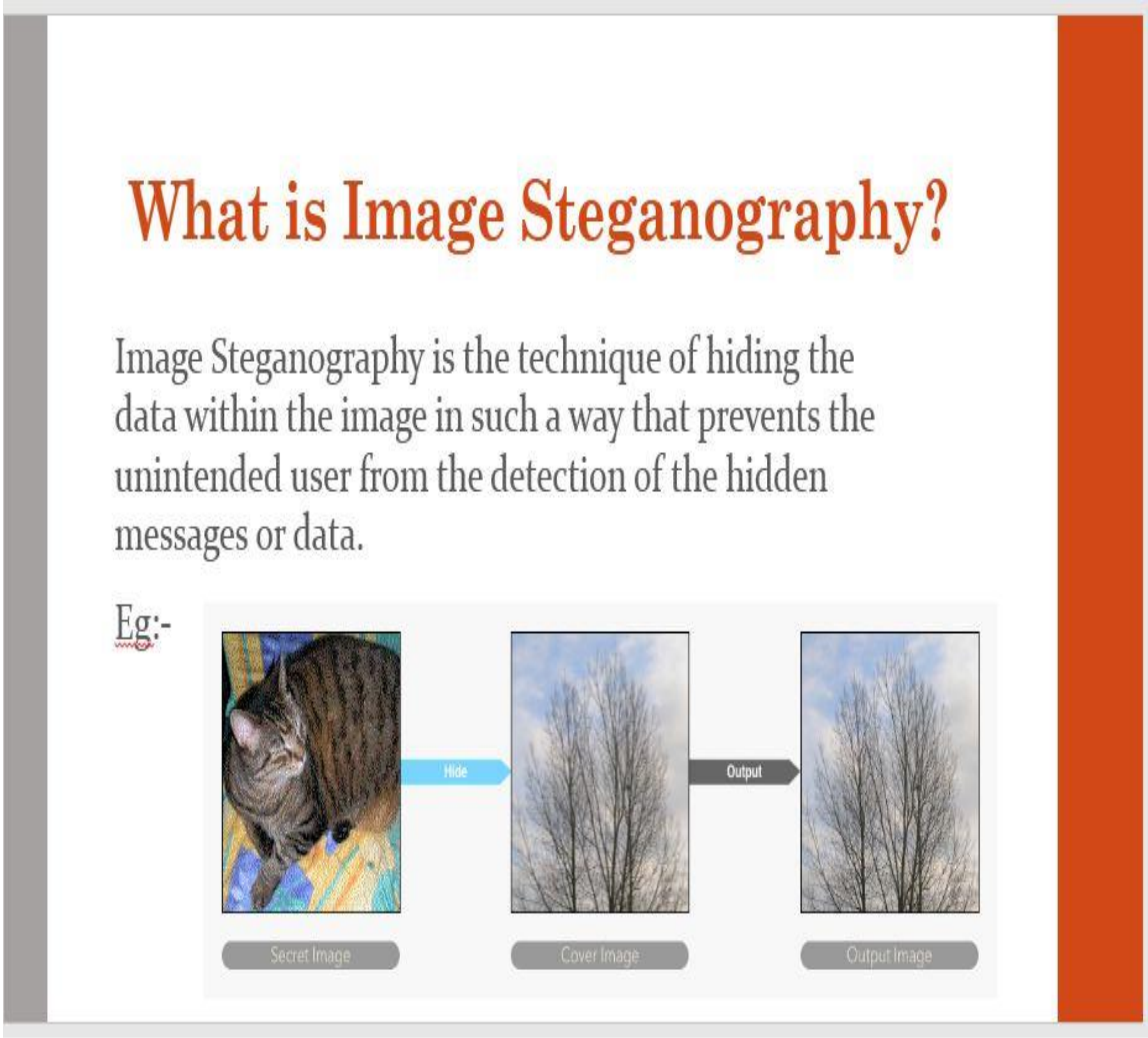
Image Steganography



What is Steganography?

Steganography is the art and science of writing hidden messages in such a way that no one, apart from the sender and intended recipient, suspects the existence of the message. Steganography can be divided into 5 types:

- Audio Steganography
- Video Steganography
- Text Steganography
- Image Steganography
- Network Steganography

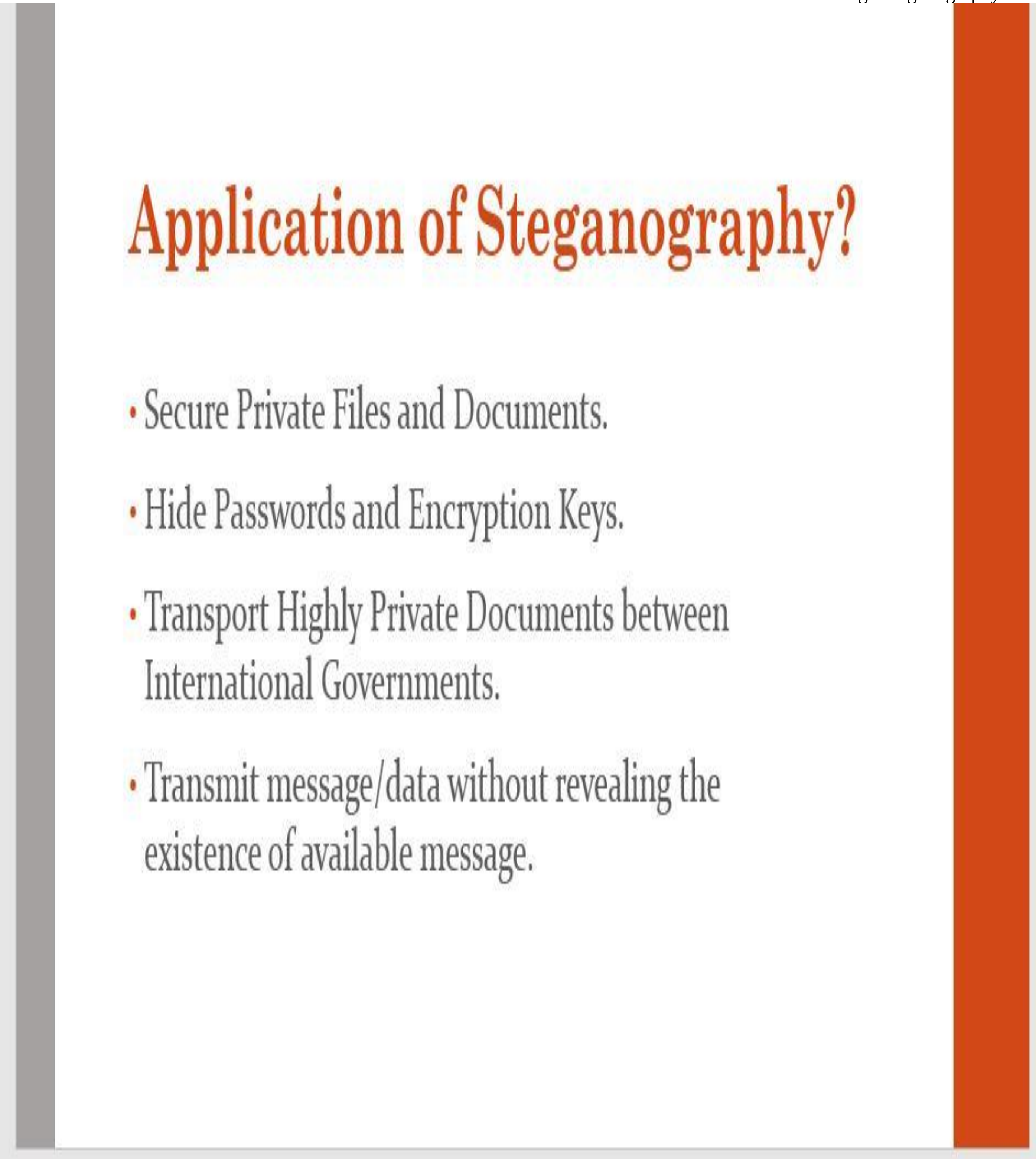


What is Image Steganography?

Image Steganography is the technique of hiding the data within the image in such a way that prevents the unintended user from the detection of the hidden messages or data.

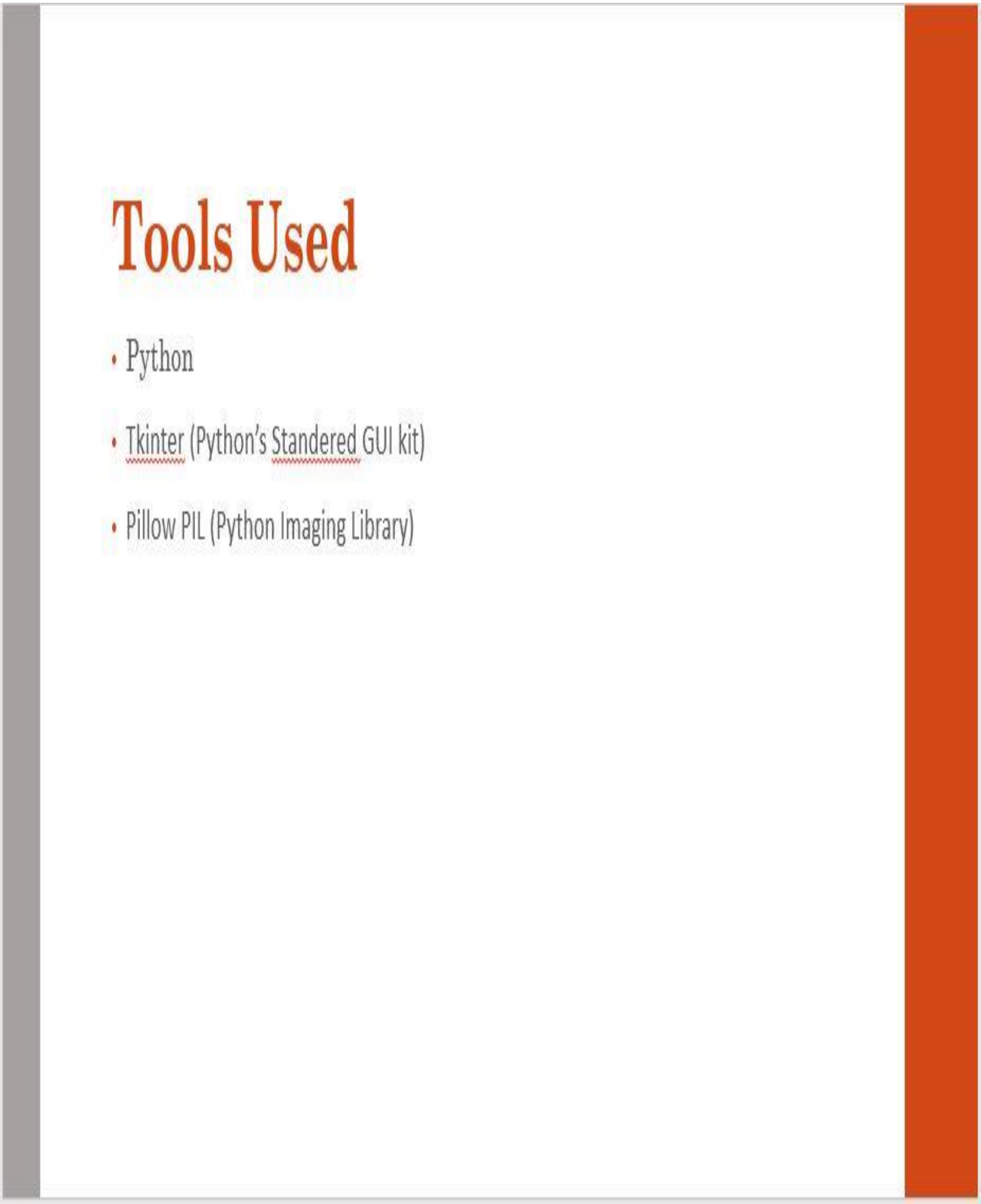
Eg:-

Hide
Output
Secret Image
Cover Image
Output Image



Application of Steganography?

- Secure Private Files and Documents.
- Hide Passwords and Encryption Keys.
- Transport Highly Private Documents between International Governments.
- Transmit message/data without revealing the existence of available message.



Tools Used

- Python
- Tkinter (Python's Standered GUI kit)
- Pillow PIL (Python Imaging Library)

THANK YOU