Title of the Project :- Stegno (Image Steganography application)

Project Description:-

Stegno (Image Steganography) is a python project in which we hide the secret message inside any image using Tkinter and the PIL module. Let's start the development.

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**.

How is it done?
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.

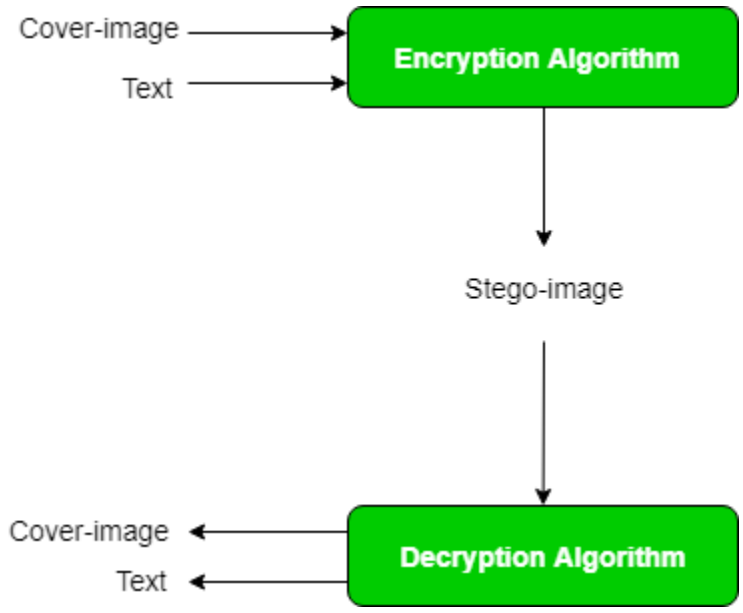


Figure – Process of Image Steganography

Is steganography a secure method of communication?

When steganography is employed alone, it is security by obscurity, which might result in the secret message being disclosed. Combining steganography and cryptography is the greatest way to disguise a message from adversaries while still protecting it in case it is detected.

In steganography, what algorithm is used?

His steganography approach entails concealing a huge amount of data (picture, audio, and text) within a colour bitmap (bmp) image. The image will be filtered and segmented in his study, with bits replacement applied to the appropriate pixels. These pixels are chosen at random rather than in order.

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.

Software Requirement :-

1. Python
2. Tkinter Library
3. PIL library

Code :-

```
from tkinter import *
from tkinter import ttk
import tkinter.filedialog
from PIL import ImageTk
from PIL import Image
from tkinter import messagebox
from io import BytesIO
import os

class Stegno:

    art ='''¯\_(ツ)_/¯'''
    art2 = '''@(\/)
            (\/)-{}-)@
            @(={}=)/\)(\/)
            (\/(/\)@| (-{}-)
            (={}=)@(\/)@(/\)@
            (/\)\(={}=)/(\/)
            @(\/)\(/\)/(={}=)
            (-{}-)"""""@/(/\)
            |: |
            /::' \\
            /::: \\
            |::' |
            |:: |
            \::. /
```

```
            ':_____.'
            `'''''''''`'''
    output_image_size = 0

    def main(self,root):
        root.title('ImageSteganography')
        root.geometry('500x600')
        root.resizable(width =False, height=False)
        f = Frame(root)

        title = Label(f,text='Image Steganography')
        title.config(font=('courier',33))
        title.grid(pady=10)

        b_encode = Button(f,text="Encode",command= lambda :self.frame1_encode(f),
padx=14)
        b_encode.config(font=('courier',14))
        b_decode = Button(f, text="Decode",padx=14,command=lambda
:self.frame1_decode(f))
        b_decode.config(font=('courier',14))
        b_decode.grid(pady = 12)

        ascii_art = Label(f,text=self.art)
        # ascii_art.config(font=('MingLiU-ExtB',50))
        ascii_art.config(font=('courier',60))

        ascii_art2 = Label(f,text=self.art2)
        # ascii_art.config(font=('MingLiU-ExtB',50))
        ascii_art2.config(font=('courier',12,'bold'))

        root.grid_rowconfigure(1, weight=1)
        root.grid_columnconfigure(0, weight=1)

        f.grid()
        title.grid(row=1)
        b_encode.grid(row=2)
        b_decode.grid(row=3)
        ascii_art.grid(row=4,pady=10)
        ascii_art2.grid(row=5,pady=5)

    def home(self,frame):
            frame.destroy()
            self.main(root)

    def frame1_decode(self,f):
        f.destroy()
```

```
        d_f2 = Frame(root)
        label_art = Label(d_f2, text='ヾ(^‿^)ﾉ')
        label_art.config(font=('courier',90))
        label_art.grid(row =1,pady=50)
        l1 = Label(d_f2, text='Select Image with Hidden text:')
        l1.config(font=('courier',18))
        l1.grid()
        bws_button = Button(d_f2, text='Select', command=lambda
:self.frame2_decode(d_f2))
        bws_button.config(font=('courier',18))
        bws_button.grid()
        back_button = Button(d_f2, text='Cancel', command=lambda :
Stegno.home(self,d_f2))
        back_button.config(font=('courier',18))
        back_button.grid(pady=15)
        back_button.grid()
        d_f2.grid()

    def frame2_decode(self,d_f2):
        d_f3 = Frame(root)
        myfile = tkinter.filedialog.askopenfilename(filetypes = ([('png', '*.png'),('jpeg',
'*.jpeg'),('jpg', '*.jpg'),('All Files', '*.*')]))
        if not myfile:
            messagebox.showerror("Error","You have selected nothing !")
        else:
            myimg = Image.open(myfile, 'r')
            myimage = myimg.resize((300, 200))
            img = ImageTk.PhotoImage(myimage)
            l4= Label(d_f3,text='Selected Image :')
            l4.config(font=('courier',18))
            l4.grid()
            panel = Label(d_f3, image=img)
            panel.image = img
            panel.grid()
            hidden_data = self.decode(myimg)
            l2 = Label(d_f3, text='Hidden data is :')
            l2.config(font=('courier',18))
            l2.grid(pady=10)
            text_area = Text(d_f3, width=50, height=10)
            text_area.insert(INSERT, hidden_data)
            text_area.configure(state='disabled')
            text_area.grid()
            back_button = Button(d_f3, text='Cancel', command= lambda :self.page3(d_f3))
            back_button.config(font=('courier',11))
            back_button.grid(pady=15)
            back_button.grid()
```

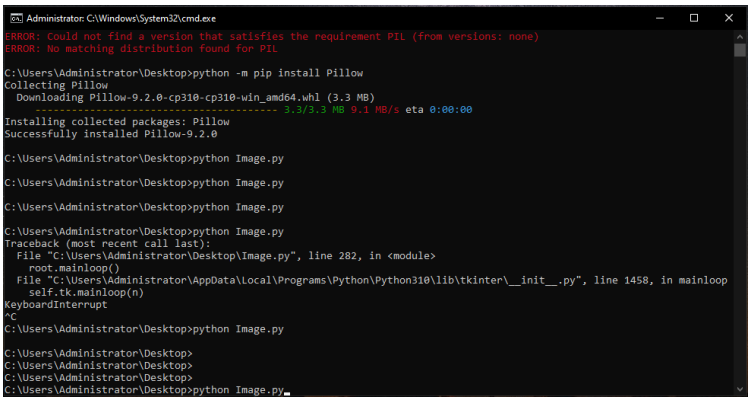
```
        show_info = Button(d_f3,text='More Info',command=self.info)
        show_info.config(font=('courier',11))
        show_info.grid()
        d_f3.grid(row=1)
        d_f2.destroy()

def decode(self, image):
    data = ''
    imgdata = iter(image.getdata())

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

def frame1_encode(self,f):
    f.destroy()
    f2 = Frame(root)
    label_art = Label(f2, text='\'\(°Ω°)/\'')
    label_art.config(font=('courier',70))
    label_art.grid(row =1,pady=50)
    l1= Label(f2,text='Select the Image in which \nyou want to hide text :')
    l1.config(font=('courier',18))
    l1.grid()

    bws_button = Button(f2,text='Select',command=lambda : self.frame2_encode(f2))
    bws_button.config(font=('courier',18))
    bws_button.grid()
    back_button = Button(f2, text='Cancel', command=lambda : Stegno.home(self,f2))
    back_button.config(font=('courier',18))
    back_button.grid(pady=15)
    back_button.grid()
    f2.grid()


def frame2_encode(self,f2):
```

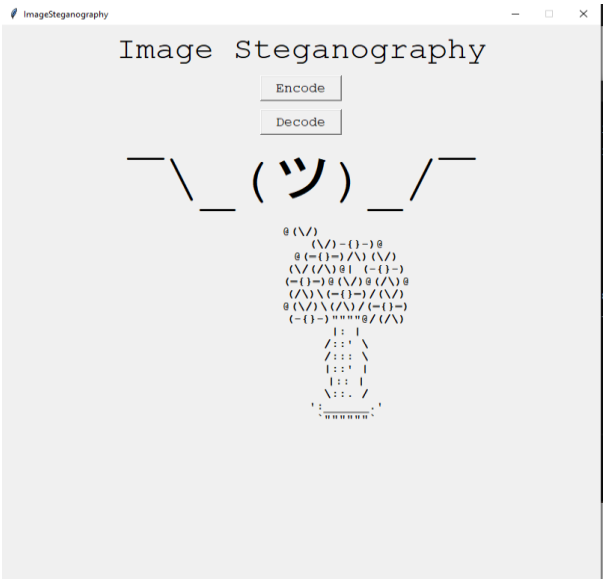
```
        ep= Frame(root)
        myfile = tkinter.filedialog.askopenfilename(filetypes = ([('png', '*.png'),('jpeg',
'*.jpeg'),('jpg', '*.jpg'),('All Files', '*.*')]))
        if not myfile:
            messagebox.showerror("Error","You have selected nothing !")
        else:
            myimg = Image.open(myfile)
            myimage = myimg.resize((300,200))
            img = ImageTk.PhotoImage(myimage)
            l3= Label(ep,text='Selected Image')
            l3.config(font=('courier',18))
            l3.grid()
            panel = Label(ep, image=img)
            panel.image = img
            self.output_image_size = os.stat(myfile)
            self.o_image_w, self.o_image_h = myimg.size
            panel.grid()
            l2 = Label(ep, text='Enter the message')
            l2.config(font=('courier',18))
            l2.grid(pady=15)
            text_area = Text(ep, width=50, height=10)
            text_area.grid()
            encode_button = Button(ep, text='Cancel', command=lambda :
Stegno.home(self,ep))
            encode_button.config(font=('courier',11))
            data = text_area.get("1.0", "end-1c")
            back_button = Button(ep, text='Encode', command=lambda :
[self.enc_fun(text_area,myimg),Stegno.home(self,ep)])
            back_button.config(font=('courier',11))
            back_button.grid(pady=15)
            encode_button.grid()
            ep.grid(row=1)
            f2.destroy()


    def info(self):
        try:
            str = 'original image:-\nsize of original image:{}mb\nwidth: {}\nheight: {}\n\n' \
                  'decoded image:-\nsize of decoded image: {}mb\nwidth: {}' \
                '\nheight: {}'.format(self.output_image_size.st_size/1000000,
                                      self.o_image_w,self.o_image_h,
                                      self.d_image_size/1000000,
                                      self.d_image_w,self.d_image_h)
            messagebox.showinfo('info',str)
        except:
            messagebox.showinfo('Info','Unable to get the information')
```

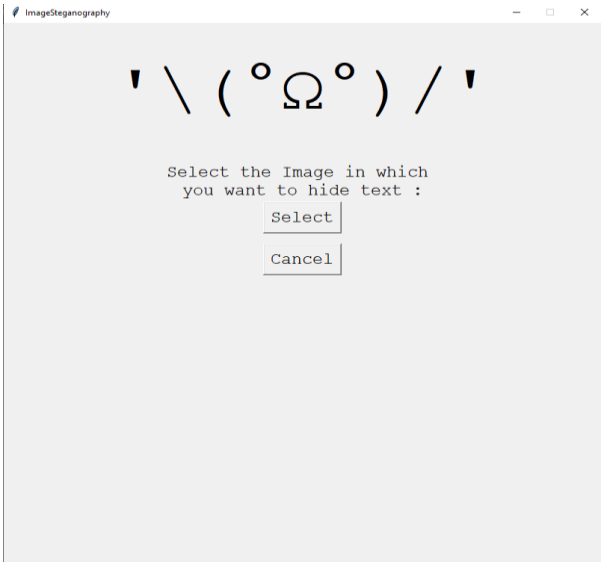
```
def genData(self,data):
    newd = []

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

def modPix(self,pix, data):
    datalist = self.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]]
        # Pixel value should be made
        # odd for 1 and even for 0
        for j in range(0, 8):
            if (datalist[i][j] == '0') and (pix[j] % 2 != 0):

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

            elif (datalist[i][j] == '1') and (pix[j] % 2 == 0):
                pix[j] -= 1
        # Eigh^th pixel of every set tells
        # whether to stop or read further.
        # 0 means keep reading; 1 means the
        # message is over.
        if (i == lendata - 1):
            if (pix[-1] % 2 == 0):
                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(self,newimg, data):
    w = newimg.size[0]
    (x, y) = (0, 0)
```

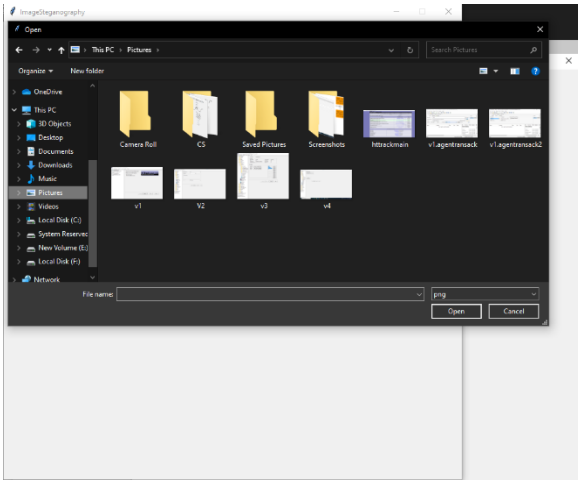
```
        for pixel in self.modPix(newimg.getdata(), data):

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

    def enc_fun(self,text_area,myimg):
        data = text_area.get("1.0", "end-1c")
        if (len(data) == 0):
            messagebox.showinfo("Alert","Kindly enter text in TextBox")
        else:
            newimg = myimg.copy()
            self.encode_enc(newimg, data)
            my_file = BytesIO()
            temp=os.path.splitext(os.path.basename(myimg.filename))[0]
            newimg.save(tkinter.filedialog.asksaveasfilename(initialfile=temp,filetypes =
([('png', '*.png')]),defaultextension=".png"))
            self.d_image_size = my_file.tell()
            self.d_image_w,self.d_image_h = newimg.size
            messagebox.showinfo("Success","Encoding Successful\nFile is saved as
Image_with_hiddentext.png in the same directory")

    def page3(self,frame):
        frame.destroy()
        self.main(root)

root = Tk()

o = Stegno()
o.main(root)

root.mainloop()
```

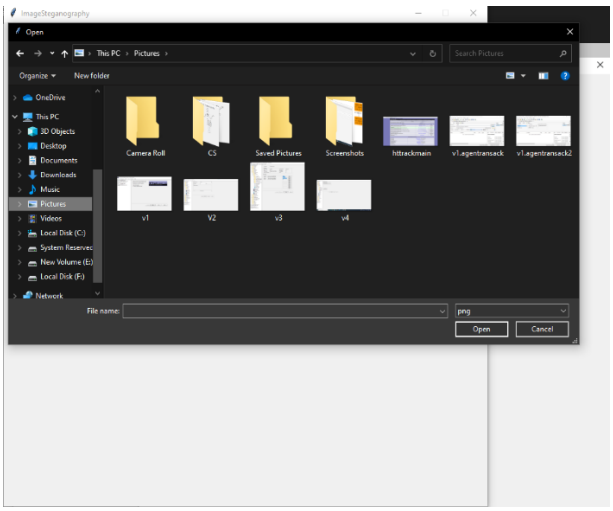
Output:

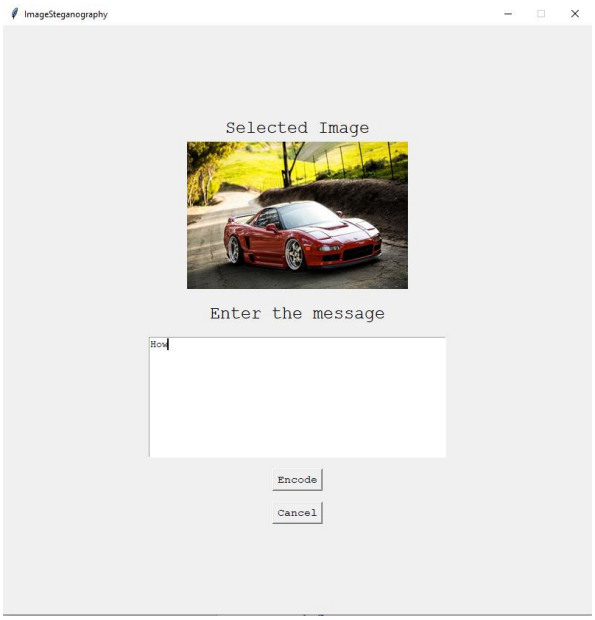

Administrator: C:\Windows\System32\cmd.exe
ERROR: Could not find a version that satisfies the requirement PIL (from versions: none)
ERROR: No matching distribution found for PIL
C:\Users\Administrator\Desktop>python -m pip install Pillow
Collecting Pillow
Downloading Pillow-9.2.0-cp310-cp310-win_amd64.whl (3.3 MB)
3.3/3.3 MB 9.1 MB/s eta 0:00:00
Installing collected packages: Pillow
Successfully installed Pillow-9.2.0
C:\Users\Administrator\Desktop>python Image.py
C:\Users\Administrator\Desktop>python Image.py
C:\Users\Administrator\Desktop>python Image.py
C:\Users\Administrator\Desktop>python Image.py
Traceback (most recent call last):
File "C:\Users\Administrator\Desktop\Image.py", line 282, in <module>
root.mainloop()
File "C:\Users\Administrator\AppData\Local\Programs\Python\Python310\lib\tkinter__init__.py", line 1458, in mainloop
self.tk.mainloop(n)
KeyboardInterrupt
^C
C:\Users\Administrator\Desktop>python Image.py
C:\Users\Administrator\Desktop>
C:\Users\Administrator\Desktop>
C:\Users\Administrator\Desktop>
C:\Users\Administrator\Desktop>python Image.py

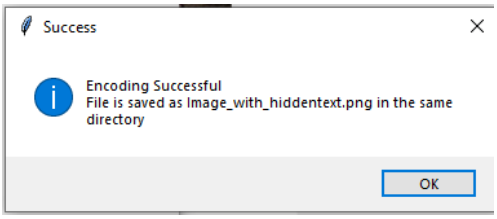

ImageSteganography
Image Steganography
Encode
Decode
¯_(ツ)_/¯

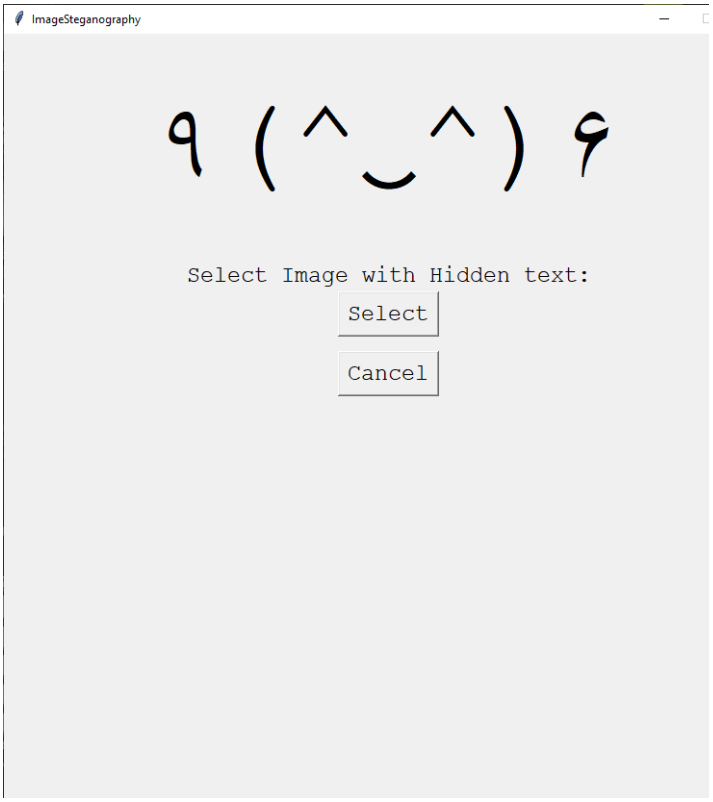

ImageSteganography
'\(°Ω°)/'
Select the Image in which
you want to hide text :
Select
Cancel

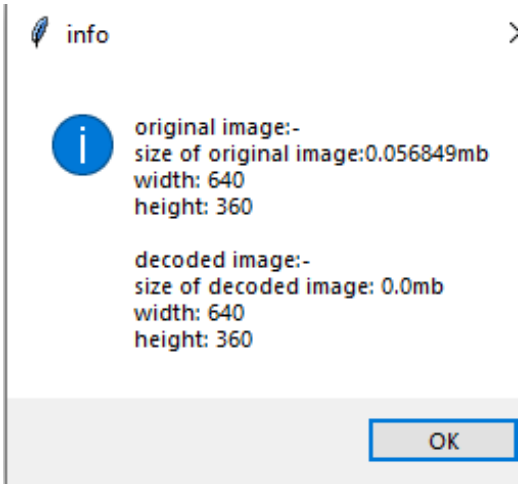

ImageSteganography
Open
This PC
Pictures
Organize
New folder
Camera Roll
CS
Saved Pictures
Screenshots
hitrackmain
v1.agentransack
v1.agentransack2
v1
V2
v3
v4
File name:
png
Open
Cancel

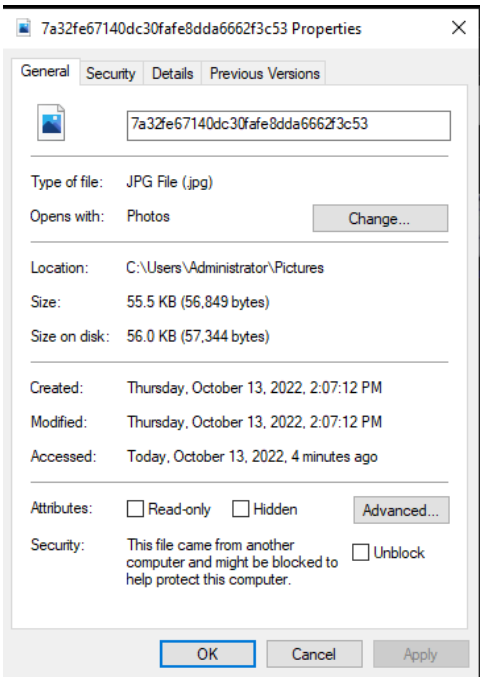

ImageSteganography
Open
This PC
Pictures
Organize
New folder
Camera Roll
CS
Saved Pictures
Screenshots
hitrackmain
v1.agentransack
v1.agentransack2
v1
V2
v3
v4
File name:
png
Open
Cancel


ImageSteganography
Selected Image
Enter the message
How
Encode
Cancel


Success
Encoding Successful
File is saved as Image_with_hiddentext.png in the same directory
OK


ImageSteganography
٩(^‿^)۶
Select Image with Hidden text:
Select
Cancel


info
original image:-
size of original image:0.056849mb
width: 640
height: 360
decoded image:-
size of decoded image: 0.0mb
width: 640
height: 360
OK



Conclusion : -

Hence We have created a project on Stegno Successfully