

Programme Name: \_\_\_\_\_\_\_\_**BCS HONS**\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

Course Code: \_\_**CSC 2730**\_\_\_\_\_\_\_

Course Name: \_\_\_\_\_\_\_\_**Network And Data Security**\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

Individual Project No. \_**1**\_\_\_

Date of Submission: \_\_\_\_\_\_**2/1/2020**\_\_\_\_\_\_\_\_\_\_\_\_\_

**Submitted By: Submitted To:**

Student Name**: Dipesh Tha Shrestha** Faculty Name**: Manoj Gautam**

IUKL ID: **041902900028** Department**: LMS**

Semester**: Third Semester**

Intake**: September 2019**

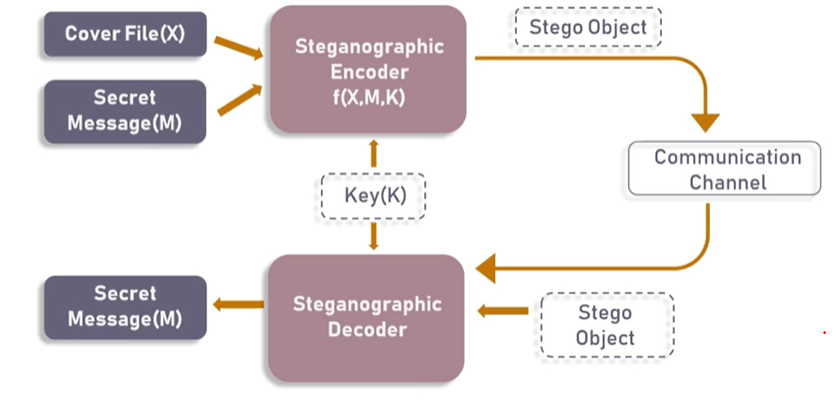
**Steganography**

Steganography is the art and science of embedding secret messages in a cover message in such a way that no one, apart from the sender and intended recipient, suspects the existence of the message

Steganography is the practice of hiding a secret message inside of (or even on top of) something that is not secret. That something can be just about anything you want. These days, many examples of steganography involve embedding a secret piece of text inside of a picture. Or hiding a secret message or script inside of a Word or Excel document.

The purpose of steganography is to conceal and deceive. It is a form of covert communication and can involve the use of any medium to hide messages. It’s not a form of cryptography, because it doesn’t involve scrambling data or using a key. Instead, it is a form of data hiding and can be executed in clever ways. Where cryptography is a science that largely enables privacy, steganography is a practice that enables secrecy – and deceit.

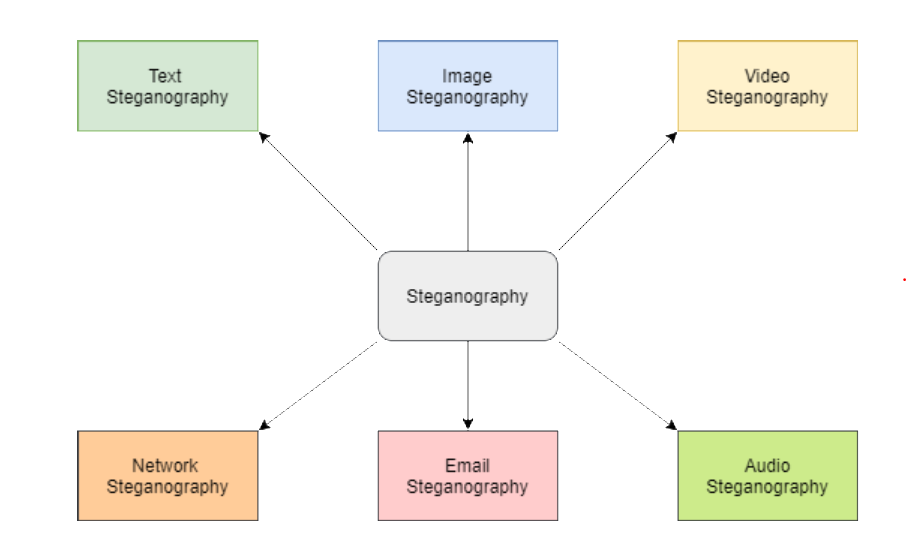
**Basic Steganographic Model**



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.

**Types of Steganography**

Steganography works have been carried out on different transmission media like images, video, text, or audio.



**Image Steganography**

* Due to the large amount of redundancy created in the manner in which digital images are represented, images are the most appropriate carrier type for steganography. Steganography on images is also the most popular form of steganography, since images occur frequently on websites, as e-mail attachments, etc. There is thus minimum cause for suspicion when a digital image is used.
* The image Steganography is used to hide a secret message inside an image. The most widely used technique to hide secret bit inside the LSB of the cover image. Because this method uses bits of each pixel in the image, it is necessary to use a lossless compression format, otherwise the hidden information will get lost in the transformations of a lossy compression algorithm.
* When using a 24 bit color image, a bit of each of the red, green and blue color components can be used, so a total of 3 bits can be used for each pixel, in this way we can use more secret bit to hide data in it.

A code for web application to securely transfer user data over the internet using image steganography. Application must have a user interface to encode and decode user message using image is given below:

|  |
| --- |
| From tkinter import \* |
|  | from tkinter.filedialog import \* |
|  | from PIL import ImageTk,Image |
|  | from tkinter import messagebox |
|  | from tkinter import messagebox as ms |
|  | from stegano import exifHeader as stg |
|  | import webbrowser |
|  | import sqlite3 |
|  |  |
|  | new = 1 |
|  | url = "https://www.gmail.com" |
|  |  |
|  | def OpenWeb(): |
|  | webbrowser.open(url,new=new) |
|  |  |
|  | with sqlite3.connect('Records.db') as db: |
|  | c = db.cursor() |
|  |  |
|  | c.execute('CREATE TABLE IF NOT EXISTS user (username TEXT NOT NULL ,password TEXT NOT NULL, Email TEXT NO NULL, Phone\_no TEXT NOT NULL);') |
|  | db.commit() |
|  | db.close() |
|  |  |
|  | def encode(): |
|  | enc=Tk() |
|  | enc.title("Encode") |
|  | enc.geometry("500x400+300+150") |
|  |  |
|  | Label(enc,text = "STEGIT", bg = "grey", width = "300", height = "3", font = ("Caliber", 13)).pack() |
|  | label1=Label(enc,text="Secret message:",bd = 3 ,font = ('',10)) |
|  | label1.place(relx=0.1,rely=0.2,height=20,width=100) |
|  |  |
|  | entry=Entry(enc,bd = 3 ,font = ('',13)) |
|  | entry.place(relx=0.4,rely=0.2) |
|  |  |
|  | label2=Label(enc,text="File name:",bd = 3 ,font = ('',10)) |
|  | label2.place(relx=0.1,rely=0.3,height=20,width=100) |
|  |  |
|  | entrysave=Entry(enc,bd = 3 ,font = ('',13)) |
|  | entrysave.place(relx=0.4,rely=0.3) |
|  |  |
|  | def openfile(): |
|  | global fileopen |
|  | fileopen=StringVar() |
|  | fileopen=askopenfilename(initialdir="/Desktop",title="select file",filetypes=(("jpeg files","\*jpg"),("all files","\*.\*"))) |
|  |  |
|  | label3=Label(enc,text=fileopen,bd = 3 ,font = ('',13)) |
|  | label3.place(relx=0.3,rely=0.4) |
|  |  |
|  |  |
|  |  |
|  | def encodee(): |
|  | response=messagebox.askyesno("pop up","do you want to encode") |
|  | if response==1: |
|  | stg.hide(fileopen,entrysave.get()+'.jpg',entry.get()) |
|  | messagebox.showinfo("pop up","successfully encode") |
|  |  |
|  | else: |
|  | messagebox.showwarning("pop up","unsuccessful") |
|  |  |
|  | buttonselect=Button(enc,text="select file",bd = 3 ,font = ('',11),command=openfile) |
|  | buttonselect.place(relx=0.1,rely=0.4) |
|  |  |
|  | buttonencode=Button(enc,text="Encode",bd = 3 ,font = ('',13),command=encodee) |
|  | buttonencode.place(relx=0.4,rely=0.6) |
|  |  |
|  |  |
|  | def decode(): |
|  | dnc=Tk() |
|  | dnc.title("Decode") |
|  | dnc.geometry("500x400+300+150") |
|  |  |
|  | def openfile(): |
|  | global fileopen |
|  | fileopen=StringVar() |
|  | fileopen=askopenfilename(initialdir="/Desktop",title="select file",filetypes=(("jpeg files","\*jpg"),("all files","\*.\*"))) |
|  |  |
|  | def decodee(): |
|  | message=stg.reveal(fileopen) |
|  |  |
|  | label4=Label(dnc,text=message,bd = 3 ,font = ('',13)) |
|  | label4.place(relx=0.3,rely=0.3) |
|  |  |
|  | buttonselect=Button(dnc,text="select file",bd = 3 ,font = ('',13),command=openfile) |
|  | buttonselect.place(relx=0.1,rely=0.3) |
|  |  |
|  | buttondecode=Button(dnc,text="Decode",bd = 3 ,font = ('',13),command=decodee) |
|  | buttondecode.place(relx=0.4,rely=0.5) |
|  |  |
|  |  |
|  |  |
|  | def stegit(): |
|  | stg = Tk() |
|  | stg.title("STEGIT") |
|  | stg.geometry("500x400+300+150") |
|  |  |
|  | Label(stg,text = "STEGIT", bg = "grey", width = "300", height = "3", font = ("Caliber", 13)).pack() |
|  | Label(stg,text="").pack() |
|  | Label(stg,text="").pack() |
|  | Button(stg,text="Encode",height="2",width="20", bd = 3 ,font = ('',13),command=encode).pack() |
|  | Label(stg,text="").pack() |
|  | Button(stg,text="Decode",height="2",width="20", bd = 3 ,font = ('',13),command=decode).pack() |
|  | Label(stg,text="").pack() |
|  | Button(stg,text="Share",height="2",width="20", bd = 3 ,font = ('',13),command=OpenWeb).pack() |
|  |  |
|  |  |
|  | class main: |
|  |  |
|  | def \_\_init\_\_(self,master): |
|  | self.master = master |
|  | self.username = StringVar() |
|  | self.password = StringVar() |
|  | self.email = StringVar() |
|  | self.Phone\_no = StringVar() |
|  | self.n\_username = StringVar() |
|  | self.n\_password = StringVar() |
|  | self.n\_email = StringVar() |
|  | self.n\_Phone\_no = StringVar() |
|  | self.widgets() |
|  |  |
|  | def login(self): |
|  | with sqlite3.connect('Records.db') as db: |
|  | c = db.cursor() |
|  |  |
|  | find\_user = ('SELECT \* FROM user WHERE username = ? and password = ?') |
|  | c.execute(find\_user,[(self.username.get()),(self.password.get())]) |
|  | result = c.fetchall() |
|  |  |
|  |  |
|  | if result: |
|  | self.logf.pack\_forget() |
|  | Label(text = "Login Successful !", width = "30", height = "3", font = ("Caliber", 13)).pack() |
|  | Button(text="OK",height="2",width="10", bd = 3 ,font = ('',13),command=stegit).pack() |
|  | Label(text="").pack() |
|  |  |
|  |  |
|  | else: |
|  | ms.showerror('Oops!','Username Not Found.') |
|  |  |
|  |  |
|  | def new\_user(self): |
|  | with sqlite3.connect('Records.db') as db: |
|  | c = db.cursor() |
|  |  |
|  | find\_user = ('SELECT \* FROM user WHERE username = ?') |
|  | c.execute(find\_user,[(self.username.get())]) |
|  | if c.fetchall(): |
|  | ms.showerror('Error!','Username Taken Try a Diffrent One.') |
|  | else: |
|  | ms.showinfo('Success!','Account Created!') |
|  | self.log() |
|  | insert = 'INSERT INTO user(username,password,email,Phone\_no) VALUES(?,?,?,?)' |
|  | c.execute(insert,[(self.n\_username.get()),(self.n\_password.get()),(self.n\_email.get()),(self.n\_Phone\_no.get())]) |
|  | db.commit() |
|  |  |
|  | def log(self): |
|  | self.username.set('') |
|  | self.password.set('') |
|  | self.email.set('') |
|  | self.Phone\_no.set('') |
|  | self.crf.pack\_forget() |
|  | self.head['text'] = 'STEGIT' |
|  | self.logf.pack() |
|  | def cr(self): |
|  | self.n\_username.set('') |
|  | self.n\_password.set('') |
|  | self.logf.pack\_forget() |
|  | self.head['text'] = 'Create Account' |
|  | self.crf.pack() |
|  |  |
|  | def widgets(self): |
|  | self.head = Label(self.master,text = 'STEGIT',font = ('',35),pady = 10) |
|  | self.head.pack() |
|  | self.logf = Frame(self.master,padx =10,pady = 20) |
|  | Label(self.logf,text = 'Username: ',font = ('',20),pady=5,padx=5).grid(sticky = W) |
|  | Entry(self.logf,textvariable = self.username,bd = 5,font = ('',15)).grid(row=0,column=1) |
|  | Label(self.logf,text = 'Password: ',font = ('',20),pady=5,padx=5).grid(sticky = W) |
|  | Entry(self.logf,textvariable = self.password,bd = 5,font = ('',15),show = '\*').grid(row=1,column=1) |
|  | Button(self.logf,text = ' Login ',bd = 3 ,font = ('',15),padx=5,pady=5,command=self.login).grid() |
|  | Button(self.logf,text = ' Register ',bd = 3 ,font = ('',15),padx=5,pady=5,command=self.cr).grid(row=2,column=1) |
|  | self.logf.pack() |
|  |  |
|  | self.crf = Frame(self.master,padx =40,pady = 50) |
|  | Label(self.crf,text = 'Username: ',font = ('',20),pady=5,padx=5).grid(sticky = W) |
|  | Entry(self.crf,textvariable = self.n\_username,bd = 4,font = ('',16)).grid(row=0,column=1) |
|  | Label(self.crf,text = 'Password: ',font = ('',20),pady=5,padx=5).grid(sticky = W) |
|  | Entry(self.crf,textvariable = self.n\_password,bd = 5,font = ('',16),show = '\*').grid(row=1,column=1) |
|  | Label(self.crf,text = 'Email: ',font = ('',20),pady=5,padx=5).grid(sticky = W) |
|  | Entry(self.crf,textvariable = self.n\_email,bd = 4,font = ('',16)).grid(row=2,column=1) |
|  | Label(self.crf,text = 'Phone no: ',font = ('',20),pady=5,padx=5).grid(sticky = W) |
|  | Entry(self.crf,textvariable = self.n\_Phone\_no,bd = 5,font = ('',16)).grid(row=3,column=1) |
|  | Label(self.crf,text="").grid() |
|  | Button(self.crf,text = 'Register',bd = 3 ,font = ('',15),padx=5,pady=10,command=self.new\_user).grid() |
|  | Button(self.crf,text = 'Go to Login',bd = 3 ,font = ('',15),padx=5,pady=10,command=self.log).grid(row=5,column=1) |
|  |  |
|  |  |
|  |  |
|  | root = Tk() |
|  | main(root) |
|  | root.mainloop() |