Image Stegnography

Submitted in partial fulfilment of the requirements for the award of the degree of

Minor Project Report

BACHELOR OF TECHNOLOGY

(Information Technology)

Submitted By:-

Vanshika Chaudhary
University Roll no.:-1905412
Class Roll no.:-1921112

GURU NANAK DEV ENGINEERING COLLEGE
LUDHIANA-141006, INDIA

Acknowledgement

We are highly grateful to Dr. Sehijpal Singh, Principal, Guru Nanak Dev Engineering College (GNDEC), Ludhiana, for providing this opportunity to carry out the minor project work at making ‘Image Stegnography ’ using unity. The constant guidance and encouragement received from Dr. K.S. Mann, H.O.D., IT Department, GNDEC Ludhiana has been of great help in carrying out the project work and is acknowledged with reverential thanks. We would like to express a deep sense of gratitude and thanks profusely to Pf. Mohanjit Kaur , without his wise counsel and able guidance, it would have been impossible to complete the project in this manner. We express gratitude to other faculty member of Information Technology department of GNDEC for their intellectual support throughout the course of this work Finally, we are indebted to all whosoever have contributed in this report work.
Vanshika Chaudhary (URN 1905412)

Contents

1 Introduction

1.1 Introduction to Project

One of the reasons that intruders can be successful is the most of the information they acquire from a system is in a form that they can read and comprehend. Intruders may reveal the information to others, modify it to misrepresent an individual or organization, or use it to launch an attack. One solution to this problem is, through the use of steganography. Steganography is a technique of hiding information in digital media. In contrast to cryptography, it is not to keep others from knowing the hidden information but it is to keep others from thinking that the information even exists. Steganography become more important as more people join the cyberspace revolution. Steganography is the art of concealing information in ways that prevents the detection of hidden messages. Steganography include an array of secret communication methods that hide the message from being seen or discovered. Steganography is of different types:
1. Text steganography
2. Image steganography
3. Audio steganography
4. Video steganography

In all of these methods, the basic principle of steganography is that a secret message is to be embedded in another cover object which may not be of any significance in such a way that the encrypted data would finally display only the cover data. So it cannot be detected easily to be containing hidden information unless proper decryption is used

As the above explanation goes, every steganography consists of three components:
1. Cover object
2. Message object
3. Resulting Steganographic object

Information hiding is an emerging research area, which encompasses applications such as copyright protection for digital media, watermarking, fingerprinting, and steganography. In watermarking applications, the message contains information such as owner identification and a digital time stamp, which usually applied for copyright protection. Fingerprint, the owner of the data set embeds a serial number that uniquely identifies the user of the data set. This adds to copyright information to makes it possible to trace any unauthorized used of the data set back to the user. Steganography hide the secrete message within the host data set and presence imperceptible and is to be reliably communicated to a receiver. The host data set is

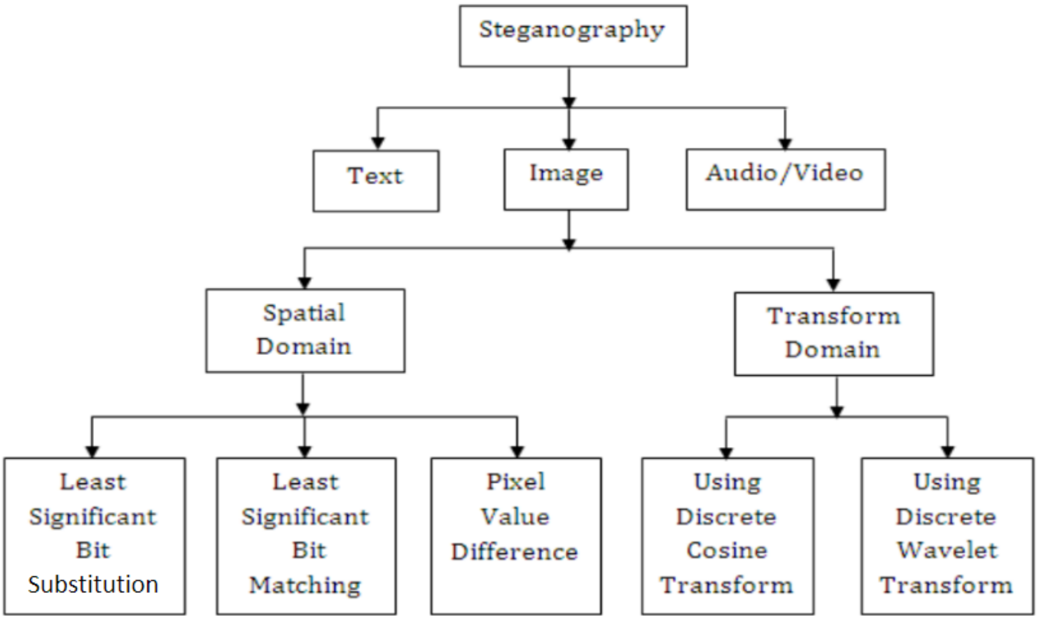

Steganography
Text
Image
Audio/Video
Spatial Domain
Transform Domain
Least Significant Bit Substitution
Least Significant Bit Matching
Pixel Value Difference
Using Discrete Cosine Transform
Using Discrete Wavelet Transform

Figure 1: Block Diagram

purposely corrupted, but in a covert way, designed to be invisible to an information analysis.

1.2 Project Category (Internet based, Application or System Development, Research based, Industry Automation, Network or System Administration)

Our project category is internet based image stegnography

1.3 Objectives

To product security tool based on stegnography techniques.
To explore techniques of hiding data using encryption model of this project.
To extract techniques to getting secret data using decryption techniques.

1.4 Problem Formulation

The former consists of linguistic or language forms of hidden writing. The later, such as invisible ink, try of hide messages physically. One disadvantage of linguistic steganography is that users must equip themselves to have a good knowledge of linguistry. In recent years, everything is trending toward digitization. And with the development of the internet technology, digital media can be transmitted conveniently over the network. Therefore, messages can be secretly carried by digital media by using the steganography techniques, and then be transmitted through the internet rapidly Steganography is the art of hiding the fact that communication is taking place, by hiding information in other information. Many different carrier file formats can be used, but digital images are the most popular because of their frequency on the internet. For hiding secret information in images, there exists a large variety of steganography techniques some are more complex than others and all of them have respective strong and weak points. So we prepare this model, to make the information hiding more simple and user friendly.

1.5 Identification/Reorganization of Need

T secure the confidential information from intruders, hackers, third party. This model helps in police government, in medical, militaries. The main reason of this model to protect data.

1.6 Existing System

In this, we are using image like jpeg, png and then write a secret message with secret key which is only visible to receiver with that key.But we have trained our model on algorithms so that it will possible outcomes in receiver as well as sender sides.

2 Requirement Analysis and System Specification

2.1 Feasibility study (Technical, Economical, Operational)

Economical: Images, code and algorithms for training the model follow all the free libraries available. No cost will be charged in developing and deploying this project. All the coding part will also be done on freely available software and editors. The cost of hardware and software will be zero, and it can be performed on any operating system. Therefore, from these details, we can say this project is economically feasible.

Technical: The image will be obtained from browser or any other software. No use of enterprise images will be needed here as only an open source will be used. Framework will be used for. The Data Integration aspect of the data manipulation will be taken care of with the use of data frame libraries such as tkinter. The primary language used to code out the exploration and analysis of data will be Python. Data analysis and exploration will be done using python libraries - tkinter and pil. pylance and gui. libraries will be utilized. Thus, this project is also technically feasible.

Operational: Operational feasibility is dependent on human resources available for the project and involves projecting whether the system will be used if it is developed and implemented. This project can be deployed on any specific domain after allotment.

2.2 Software Requirement Specification Document

Data Requirement- Data required for this project is available online on a site. Data is required for this project for training of the model. For training the dataset was divided into training and testing dataset in the ratio of 80-20.

Functional Requirements- Functional requirements in an SRS document (software requirements specification) indicate what a software system must do and how it must function; they are product features that focus on user needs. Our model will function as it is a encryption decryption techniques to hide confidential data.

Performance Requirement- System performance is the most important quality in non functional requirements and affects almost all the other preceding ones. Our model can produce the output of the data of image and about the private key in just a few milliseconds and helps to give suggestion.

Dependability Requirement- Dependability is the probability and percentage of the software performing without failure for a specific number of uses or amount of time. This feature defines the amount of time the system is running, the time it takes to repair a fault, and the time between lapses. Users can get their output of the desired one.

Maintainability requirement- Maintainability defines the time required for a solution or its component to be fixed, changed to increase performance or other qualities, or adapted to a changing environment. If our model's prediction become unavailable, they can be under maintenance for approximately one or two hours. This feature is defined as the ability to control a system efficiently and keep it fully operational.

Security requirement- Security measures ensure your software's safety againstespionage or sabotage. These features are necessary even for stand-alone systems; you don't want anyone to have access to your sensitive data. As we had deployed on vscode, and anyone can access it but no one can change the backend part so it is fully secured.

Look and feel requirement- No external requirements will affect our model. It will just run on the backend irrespective of the environment conditions. All system components must follow a common and standard set of exchange formats to exchange data; the lack of interoperability happens when people do not follow standards. Feel indicates how effectively they can learn and use a system.

2.3 Validation

This model will validate the art of concealing the fact that connection is taking place, by hiding data in other information. Some different carrier file structure can be used, but digital images are the important because of their frequency on the computer network.

2.4 Expected hurdles

While making this model we came across many hurdles, some of them are listed below: First, it was difficult to put the correct algorithm technique like there are

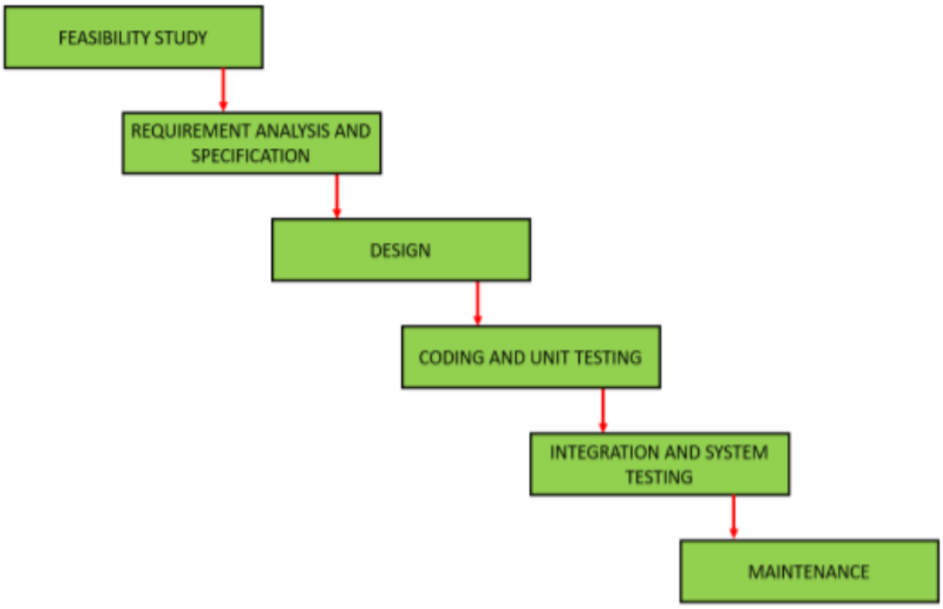

FEASIBILITY STUDY
REQUIREMENT ANALYSIS AND SPECIFICATION
DESIGN
CODING AND UNIT TESTING
INTEGRATION AND SYSTEM TESTING
MAINTENANCE

Figure 2: Waterfall model flowchart

diff algorithm RSA, LSB, Diffie helman. On this basis, this was quite difficult to implement this, in our code.

2.5 SDLC model to be used

We used a waterfall model in making this project.

Requirement Gathering and analysis - In this phase we have collected resarch work, code, where to implement.

System Design - We have been using different algorithm and code for the design of the system and then compared all the algorithm based . Implementation We implemented the model using like vscode, online editor, anaconda, android studio.

Integration and Testing - After that we have tested our model on vscode . Deployment of system For the deployment phase we have integrated on vscode, with image, secrect message, and key.

Maintenance - The maintenance part is also easy as it is easy to edit code part because we had used proper comments to tell what we want to do in that part of the code section.

3 System Design

3.1 Introduction

The system design is the most creative and challenging phase of the system development life cycle. It is an approach for the creation of the proposed system, which still helps the system coding. It provides the understanding and procedural details necessary for implementing the system. A number of subsystems are to be identified which constitutes the whole system. In this phase the data organization is to be discussed, in which the output formats are to be designed. The system design is composed of several steps.
Here the emphasis is on translating the performance requirements in to the design specification. Steganography system requires any type of image file and the information or message that is to be hidden. It has two modules encrypt and decrypt. Microsoft .Net framework prepares a huge amount of tool and options for programmers that they simples programming.

One of .Net tools for pictures and images is auto-converting most types of pictures to BMP format. I used this tool in this software called "Steganography" that is written in C.Net language and you can use this software to hide your information in any type of pictures without any converting its format to BMP (software converts inside it).

The algorithm used for Encryption and Decryption in this application provides using several layers lieu of using only LSB layer of image. Writing data starts from last layer (8st or LSB layer); because significant of this layer is least and every upper layer has doubled significant from its down layer. So every step we go to upper layer image quality decreases and image retouching transpires.

The encrypt module is used to hide information into the image; no one can see that information or file. This module requires any type of image and message and gives the only one image file in destination.

The decrypt module is used to get the hidden information in an image file. It take the image file as an output, and give two file at destination folder, one is the same image file and another is the message file that is hidden it that.

Before encrypting file inside image we must save name and size of file in a definite place of image. We could save file name before file information in LSB layer and save file size and file name size in most right-down pixels of image. Writing this

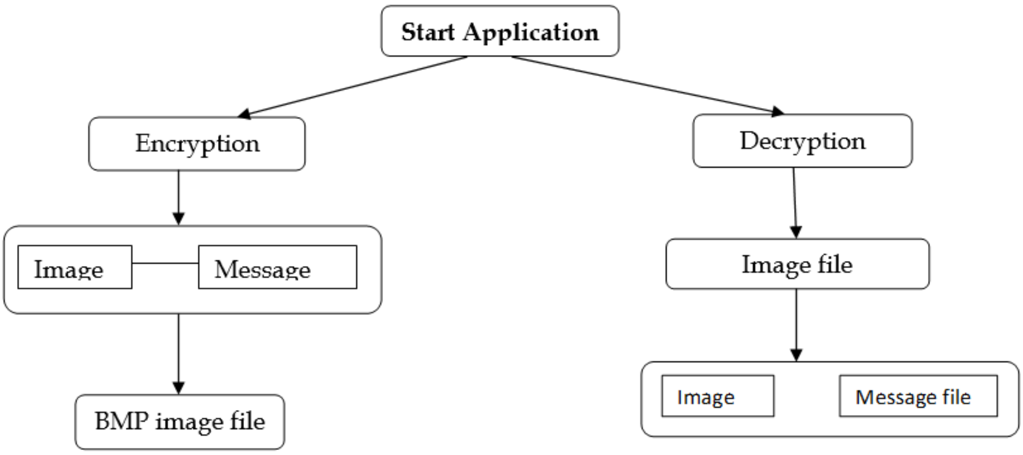


Figure 3: Block diagram

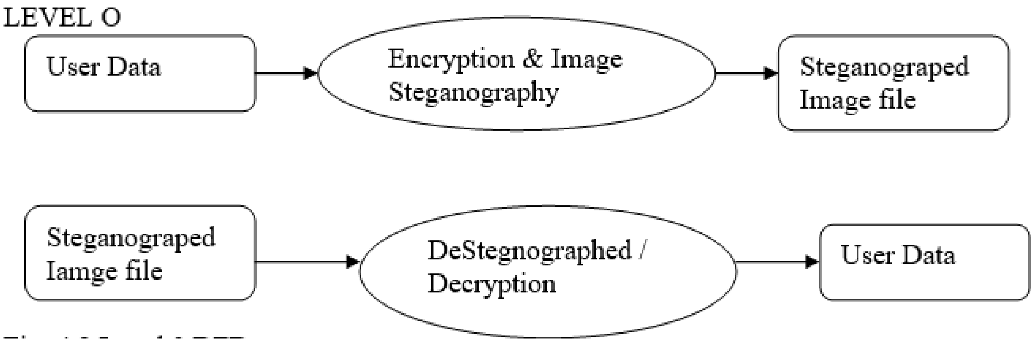


Figure 4: Block diagram

information is needed to retrieve file from encrypted image in decryption state. The graphical representation of this system is as follows:

LEVEL 1

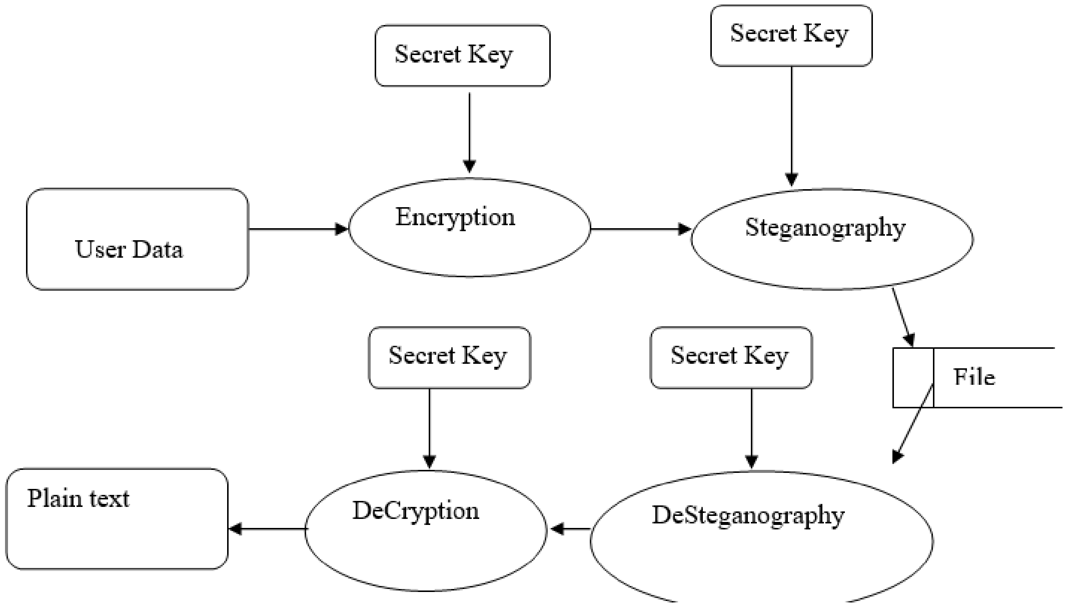


Fig 3.3 Block diagram

The level 0 of DFD specifies the user data that is encrypted and then the cipher text is hiding into the image file using steganography. The output obtained is steganographed audio file. (sing steganographic algorithm the encrypted text is separated from the image file. (sing decryption the cipher text is converted to user data.The level 1 of DFD specifies the user data which is encrypted using secret key and then the cipher text is hiding into the image file using steganography with the help of secret key. Thus the image file with encrypted data is stored in a file. The image file is steganographed with the help of secret key. Thus the encrypted text is separated from image file. (sing decryption cipher text is converted to plain text using secret key

4 Implementation

Implementation is one of the most important tasks in project implementation is the phase, in which one has to be cautious, because all the efforts undertaken during the project will be fruitful only if the software is properly implemented according to the plans made. The implementation phase is less creative than system design. It is primarily concerned with user-training, site preparation and file-sites, the tests of the network along with the systems are also included under implementation. Depending on the nature of the systems extensive user training may be required. Programming is itself a design works. The initial parameters of the management information systems should be modified as a result of programming efforts. Programming provides a real test for the assumption made by the analyst. System testing checks the readiness and accuracy of the system to access, update and retrieve data from the new files. Once the programs become available the test data are read into the computer and processed. In most conventions parallel run was conducted to establish the efficiency of the system. Implementation is used here to mean the process of converting a new or revised system in to an operational one. Conversion is one aspect of one implementation.

5 Testing

Testing is the process of evaluating a system by manual or automatic means to verify that it satisfies the specified requirements or to identify differences between the actual and expected results. During system testing, the system is used experimentally to ensure that the software does not fail. special test data are input for processing and the results are examined. If the program fails to behave as expected, then the conditions under which such a failure occurs are noted for debugging and correction. program testing represents the logical elements of the system. The testing ultimately leads to suffice the quality factors such as correctness, reliability, efficiency, usability, maintainability, portability,accuracy, error tolerance and expand ability.

5.1 Different Types of Testing

- Test Strategies -
  For testing software, various test strategies can be used such as Unit Testing Integration Testing, Black-box testing, White-box testing, regression testing, and acceptance testing etc.

- Unit Testing -
  Unit test focuses verification effort on the smallest unit of software design, the software components or modules. By testing in this method we should be very clear of the bugs occured. The module interfaces is tested to ensure that information property flows into and out of the program under test.

- Integration Testing -
  Integration testing is a systematic technique for constructing the program structure while at the same time conducting tests to uncover errors associated with interfacing. The objective is to take unit tested components and build a program structure that has been dedicated by design.

- Black-box Testing:
  Black box testing alludes to test that are conducted at the software interface. Although they are designed to uncover errors, black-box tests are used to demonstrate that software functions are operational, that input is properly accepted and the output is correctly produced. A black-box test examines some fundamental aspect of a system with title regard for the internal logical structure of the software.

- White box testing is predicated on close examination of procedural detail. Logical paths through software are tested by providing test cases that exercise specific set of conditions and loops.

- Regression Testing:
  Regression testing involves executing old test cases to test that no new errors have been introduced. This testing is performed when some changes are made to an existing system. The modified system needs to be tested to make sure that the new features to be added to work.

- Acceptance Testing:
  Acceptance testing involves planning and execution of functional tests, performance test to verify that implemented system satisfies its requirements. Acceptance tests are typically performed by the quality assurance and customer organization.

5.2 Preconditions

The following items are required before testing can take place:

- A complete and coherent functional specification of the Steganography System expressed as use cases and usage scenarios.
- A complete and validation-tested release of Steganography System, delivered according to the delivery plan.
- An agreed-upon procedure for dealing with any anomalies that are discovered during the testing process.
- A set of test specifications describing how each functional area of the Steganography System is to be acceptance tested.
- An implemented test environment for the testing Sufficient, suitable resources to carry out the testing.
- Available standards for the acceptance testing.

5.3 Test Priorities

During testing of the Steganography System, the following qualities will be tested in order of priority:

- Functionality—whether the required functions are available and working as expected.

- Usability—how user-friendly and intuitive the Steganography System is
- Security—how well-protected and guaranteed corporate and user data is
- Performance—whether the response times are within acceptable limits
- Customization—how straightforward it is to use the application in new, unpredicted ways

5.4 Test Organization

5.4.1 Roles And Responsibilities

The following roles are defined:

- QA lead/test manager—responsible for planning and ensuring the smooth running of the test process
- Tester—carries out the tests according to the test plan, and then reports the results
- Product manager—ensures that the tests are carried out successfully from a user perspective
- Project sponsor/client—acts as main stakeholder, and ensures that the needs of the customer community as a whole are considered
- Test support—provides technical assistance, such as test environment configuration, and non-technical assistance, such as methodological support.

Weekly team meetings will be held involving the test manager, testers, and product managers. At these meetings, the progress of the testing process will be reported, any issues will be discussed, and actions will be agreed upon.

5.5 Test Environment

5.5.1 Hardware And Software

The test environment will consist of:

Server

1. Microsoft Windows
2. Steganography System components
3. One laser printer to print reports
4. One color printer (laser or inkjet) to print screen dumps
5. One CD-ROM drive to enable clean installation of the Steganography System

5.6 Application Configuration

The following user accounts will be configured on the server:
System User 1
System User 2

5.7 Test Management

Tests shall be managed according to the corporate test management standards, which cover:

1. Conduct of tests
2. Reporting of test results
3. Defect tracking and resolution
4. Configuration management of the test environment
5. Configuration control of test deliverable.

5.8 Description

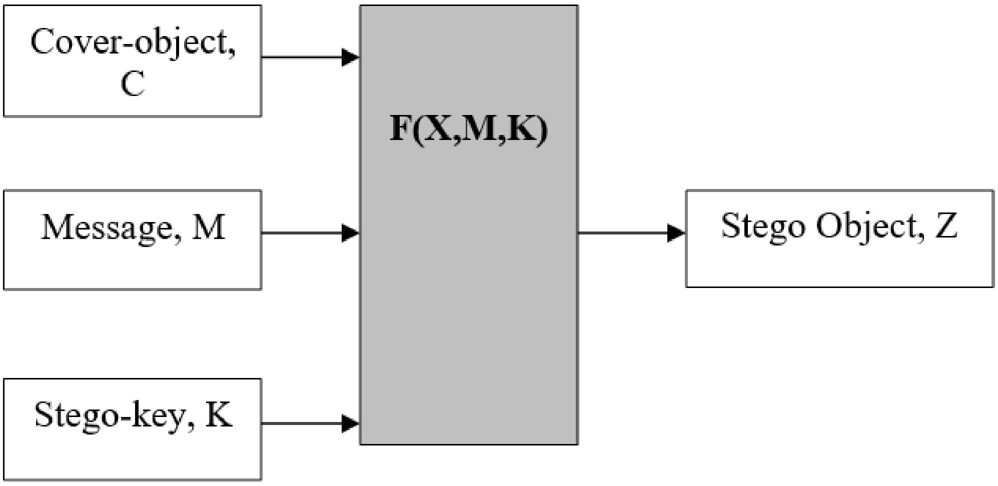
User browse image file and data file to hide data file into image file and save new bitmap file in encryption option, User browse encrypted bitmap file and decrypt the file to get hidden data file by using decrypt option.
Test Actions

- Click Encrypt button
- Browse the Bitmap file.
- Browse the data file.
- Click on Encrypt button.
- System save the new encrypted file
- Click on Decrypt button
- Browse encrypted bitmap file.
- Browse folder to get hidden data
- Click on Decrypt button to get the hidden data file

6 Critical Evalution

The former consists of linguistic or language forms of hidden writing. The later, such as invisible ink, try of hide messages physically. One disadvantage of linguistic steganography is that users must equip themselves to have a good knowledge of inguistry. In recent years, everything is trending toward digitization. And with the development of the internet technology, digital media can be transmitted conveniently over the network. Therefore, messages can be secretly carried by digital media by using the steganography techniques, and then be transmitted through the internet rapidly Steganography is the art of hiding the fact that communication is taking place, by hiding information in other information. Many different carrier file formats can be used, but digital images are the most popular because of their frequency on the internet. For hiding secret information in images, there exists a large variety of steganography techniques some are more complex than others and all of them have respective strong and weak points.

So we prepare this framework, to make the information hiding more simple and user friendly. Basically, the model for steganography is shown on following figure:



Diagram

Message is the data that the sender wishes to remain it confidential. It can be plain text, ciphertext, other image, or anything that can be embedded in a bit stream such as a copyright mark, a covert communication, or a serial number. Password is known as stego-key, which ensures that only recipient who know the corresponding decoding key will be able to extract the message from a cover-object. The cover-object with the secretly embedded message is then called the Stego-object. Recovering message from a stego-object requires the cover-object itselt and a corresponding decoding key if a stego-key was used during the encoding process. The original image may or may not be required in most applications to extract the message.

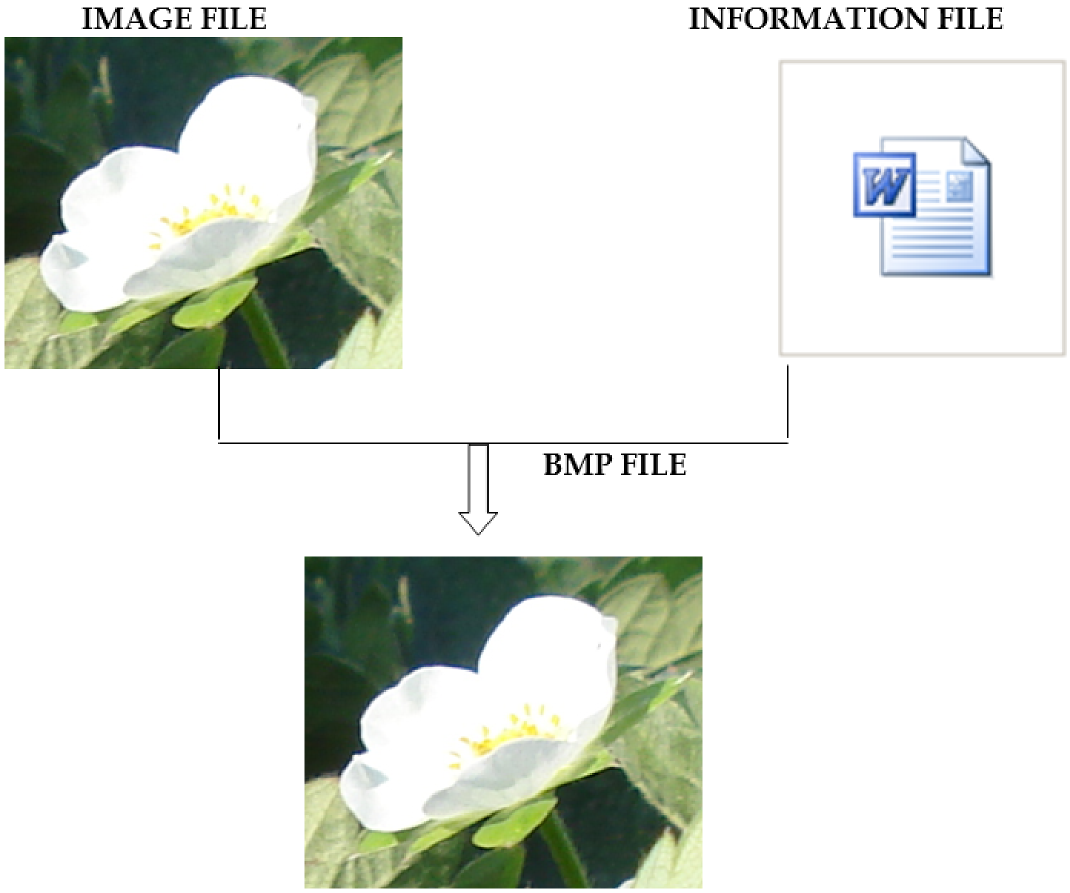


Fig 6.1 Encryption Process

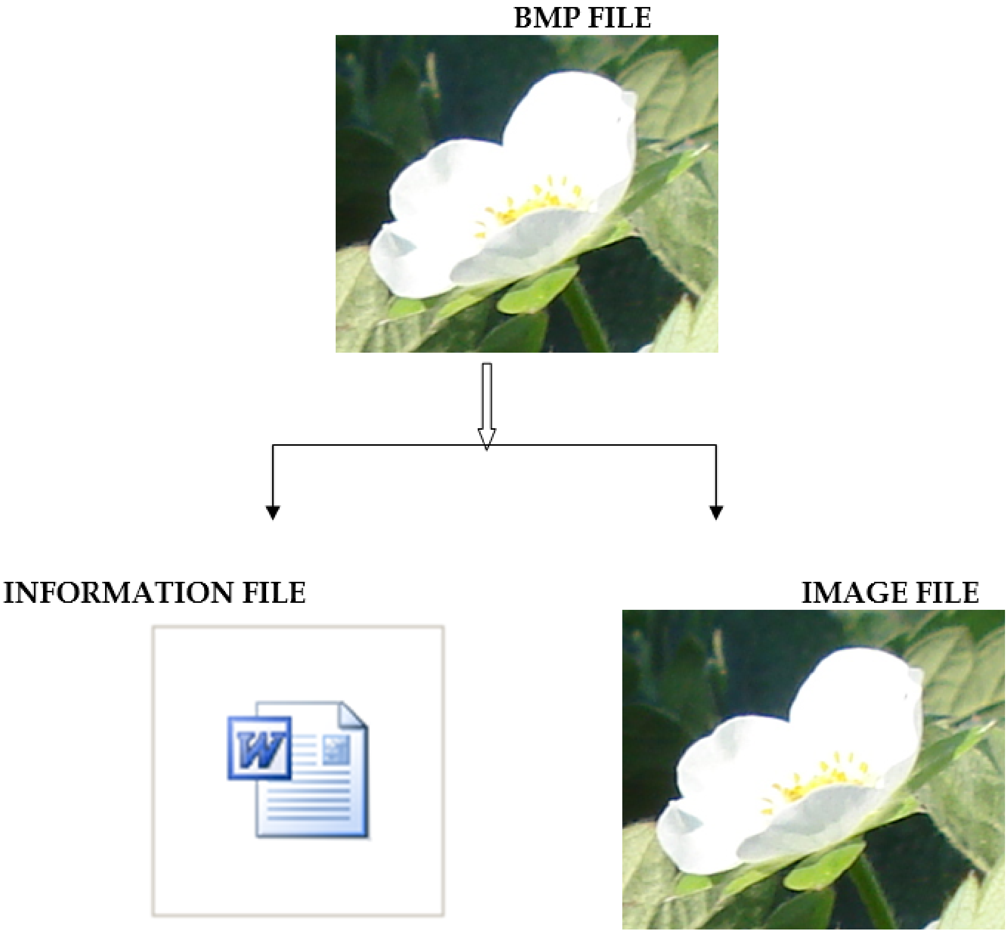


Fig 6.2 Decryption Process

7 Code Analysis

DataFlair- import modules
from tkinter import *
import tkinter.filedialog
from tkinter import messagebox
from PIL import ImageTk
from PIL import Image
from io import BytesIO
import os

class IMG$_{S}tegno :$
$output_{i}mage_{s}ize = 0$

main frame or start page
def main(self, root):
root.title('ImageSteganography by DataFlair')
root.geometry('500x600')
root.resizable(width =False, height=False)
root.config(bg = 'e3f4f1')
frame = Frame(root)
frame.grid()

title = Label(frame,text='DataFlair Image Steganography')
title.config(font=('Times new roman',25, 'bold'))
title.grid(pady=10)
title.config(bg = 'e3f4f1')
title.grid(row=1)

encode = Button(frame,text="Encode",command= lambda
:self.encode$_{f}rame1(frame), padx = 14, bg =' e3f4f1')$
$encode.config(font = ('Helvetica', 14), bg =' e8c1c7')$
$encode.grid(row = 2)$
$decode = Button(frame, text = "Decode", command = lambda$
$: self.decode_{f}rame1(frame), padx = 14, bg =' e3f4f1')$
$decode.config(font = ('Helvetica', 14), bg =' e8c1c7')$
$decode.grid(pady = 12)$
$decode.grid(row = 3)$

root.grid$_{r}owconfigure(1, weight = 1)$
$root.grid_{c}olumnconfigure(0, weight = 1)$

back function to loop back to main screen
def back(self,frame):
frame.destroy()
self.main(root)

frame for encode page
def encode$_{f}rame1(self, F):$
$F.destroy()$
$F2 = Frame(root)$
$label1 = Label(F2, text =' SelecttheImageinwhichwanttohidetext :')$
$label1.config(font = ('Timesnewroman', 25, 'bold'), bg =' e3f4f1')label1.grid()$

button$_{b}ws = Button(F2, text =' Select', command = lambda :$
$self.encode_{f}rame2(F2))$
$button_{b}ws.config(font = ('Helvetica', 18), bg =' e8c1c7')$
$button_{b}ws.grid()$
$button_{b}ack = Button(F2, text =' Cancel', command = lambda :$
$IMG_{S}tegno.back(self, F2))$
$button_{b}ack.config(font = ('Helvetica', 18), bg =' e8c1c7')$
$button_{b}ack.grid(pady = 15)$
$button_{b}ack.grid()$
$F2.grid()$

frame for decode page
def decode$_{f}rame1(self, F):$
$F.destroy()$
$d_{f}2 = Frame(root)$
$label1 = Label(d_{f}2, text =' SelectImagewithHiddentext :')$
$label1.config(font = ('Timesnewroman', 25, 'bold'), bg =' e3f4f1')$
$label1.grid()$
$label1.config(bg =' e3f4f1')$
$button_{b}ws = Button(d_{f}2, text =' Select', command = lambda$
$: self.decode_{f}rame2(d_{f}2))$
$button_{b}ws.config(font = ('Helvetica', 18), bg =' e8c1c7')$
$button_{b}ws.grid()$
$button_{b}ack = Button(d_{f}2, text =' Cancel', command = lambda :$
$IMG_{S}tegno.back(self, d_{f}2))$

$button_back.config(font = ('Helvetica', 18), bg =' e8c1c7')$
$button_back.grid(pady = 15)$
$button_back.grid()$
$d_f2.grid()$

function to encode image
def encode$_frame2(self, e_F2)$:
$e_pg = Frame(root)$
$myfile = tkinter.filedialog.askopenfilename(filetypes = ([('png',$
$'.png'), ('jpeg',' .jpeg'), ('jpg',' .jpg'), ('AllFiles',' .*')]))$
$ifnotmyfile :$
$messagebox.showerror("Error", "Youhaveselectednothing!")$
$else :$
$my_img = Image.open(myfile)$
$new_image = my_img.resize((300, 200))$
$img = ImageTk.PhotoImage(new_image)$
$label3 = Label(e_pg, text =' SelectedImage')$
$label3.config(font = ('Helvetica', 14,' bold'))$
$label3.grid()$
$board = Label(e_pg, image = img)$
$board.image = img$
$self.output_image_size = os.stat(myfile)$
$self.o_image_w, self.o_image_h = my_img.size$
$board.grid()$
$label2 = Label(e_pg, text =' Enterthemessage')$
$label2.config(font = ('Helvetica', 14,' bold'))$
$label2.grid(pady = 15)$
$text_a = Text(e_pg, width = 50, height = 10)$
$text_a.grid()$
$encode_button = Button(e_pg, text =' Cancel', command = lambda :$
$IMG_Stegno.back(self, e_pg))$
$encode_button.config(font = ('Helvetica', 14), bg =' e8c1c7')$
$data = text_a.get("1.0", "end - 1c")$
$button_back = Button(e_pg, text =' Encode', command = lambda :$
self.enc$_fun(text_a, my_img), IMG_Stegno.back(self, e_pg)$
)
$button_back.config(font = ('Helvetica', 14), bg =' e8c1c7')$
$button_back.grid(pady = 15)$
$encode_button.grid()$
$e_pg.grid(row = 1)$
$e_F2.destroy()$

function to decode image
def decode$_{f}rame2(self, d_{F}2):$
$d_F3 = Frame(root)$
$myfiles = tkinter.filedialog.askopenfilename(filetypes = ([('png',$
$'.png'),('jpeg','.jpeg'),('jpg','.jpg'),('AllFiles','.*')]))$
$ifnotmyfiles:$
$messagebox.showerror("Error","Youhaveselectednothing!")$
$else:$
$my_img = Image.open(myfiles,'r')$
$my_image = my_img.resize((300,200))$
$img = ImageTk.PhotoImage(my_image)$
$label4 = Label(d_F3, text =' SelectedImage :')$
$label4.config(font = ('Helvetica', 14,' bold'))$
$label4.grid()$
$board = Label(d_F3, image = img)$
$board.image = img$
$board.grid()$
$hidden_data = self.decode(my_img)$
$label2 = Label(d_F3, text =' Hiddendatais :')$
$label2.config(font = ('Helvetica', 14,' bold'))$
$label2.grid(pady = 10)$
$text_a = Text(d_F3, width = 50, height = 10)$
$text_a.insert(INSERT, hidden_data)$
$text_a.configure(state =' disabled')$
$text_a.grid()$
$button_back = Button(d_F3, text =' Cancel', command = lambda$
$: self.frame_3(d_F3))$
$button_back.config(font = ('Helvetica', 14), bg =' e8c1c7')$
$button_back.grid(pady = 15)$
$button_back.grid()$
$d_F3.grid(row = 1)$
$d_F2.destroy()$

function to decode data
def decode(self, image):
image$_data = iter(image.getdata())$
$data =''$

while (True):

pixels = [value for value in next(image$_{d}ata)[: 3] +$
$next(image_{d}ata)[: 3] +$
$next(image_{d}ata)[: 3]]$
$binary_{s}tr =''$
$foriinpixels[: 8] :$
$ifibinary_{s}tr+ =' 0'$
$else :$
$binary_{s}tr+ =' 1'$

data += chr(int(binary$_{s}tr, 2))$
$ifpixels[-1]returndata$

function to generate data
def generate$_{D}ata(self, data) :$
$new_{d}ata = []$

for i in data:
new$_{d}ata.append(format(ord(i),' 08b'))$
$returnnew_{d}ata$

function to modify the pixels of image
def modify$_{P}ix(self, pix, data) :$
$dataList = self.generate_{D}ata(data)$
$dataLen = len(dataList)$
$imgData = iter(pix)$
$foriinrange(dataLen) :$
$Extracting3pixelsatatime$
$pix = [valueforvalueinnext(imgData)[: 3] +$
$next(imgData)[: 3] +$
$next(imgData)[: 3]]$

for j in range(0, 8):
if (dataList[i][j] == ’0’) and (pix[j] if (pix[j] pix[j] -= 1

elif (dataList[i][j] == ’1’) and (pix[j] pix[j] -= 1

if (i == dataLen - 1):
if (pix[-1] pix[-1] -= 1
else:

if (pix[-1] pix[-1] -= 1

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

function to enter the data pixels in image
def encode$_{e}nc(self, newImg, data) :$
$w = newImg.size[0]$
$(x, y) = (0, 0)$

for pixel in self.modify$_{P}ix(newImg.getdata(), data) :$

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

function to enter hidden text
def enc$_{f}un(self, text_{a}, myImg) :$
$data = text_{a}.get("1.0", "end - 1c")$
$if(len(data) == 0) :$
$messagebox.showinfo("Alert", "KindlyentertextinTextBox")$
$else :$
$newImg = myImg.copy()$
$self.encode_{e}nc(newImg, data)$
$my_{f}ile = BytesIO()$
$temp = os.path.splitext(os.path.basename(myImg.filename))[0]$
$newImg.save(tkinter.filedialog.asksaveasfilename(initialfile = temp, filetypes =$
$([('png', '*.png')]), defaultextension = ".png"))$
$self.d_{i}mage_{s}ize = my_{f}ile.tell()$
$self.d_{i}mage_{w}, self.d_{i}mage_{h} = newImg.size$
$messagebox.showinfo("Success", "EncodingSuccessfulissaved$
$asImage_{w}ith_{h}iddentext.pnginthesamedirectory")$

def frame$_{3}(self, frame) :$

$frame.destroy()$
$self.main(root)$

GUI loop
root = Tk()
o = IMG$_Stegno()$
$o.main(root)$
$root.mainloop()$

8 Result

(a) To start the model, we have to click on the encode option

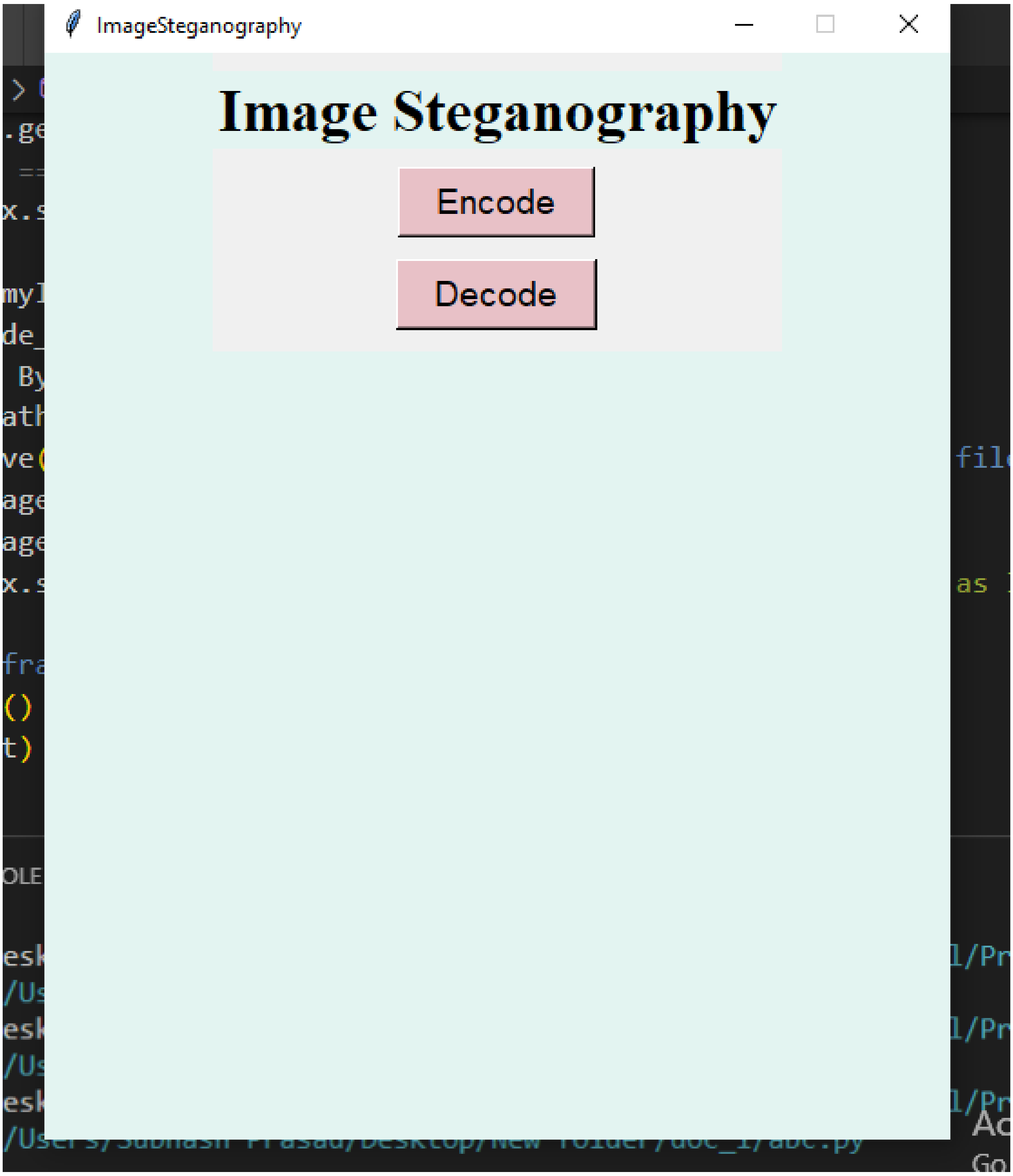
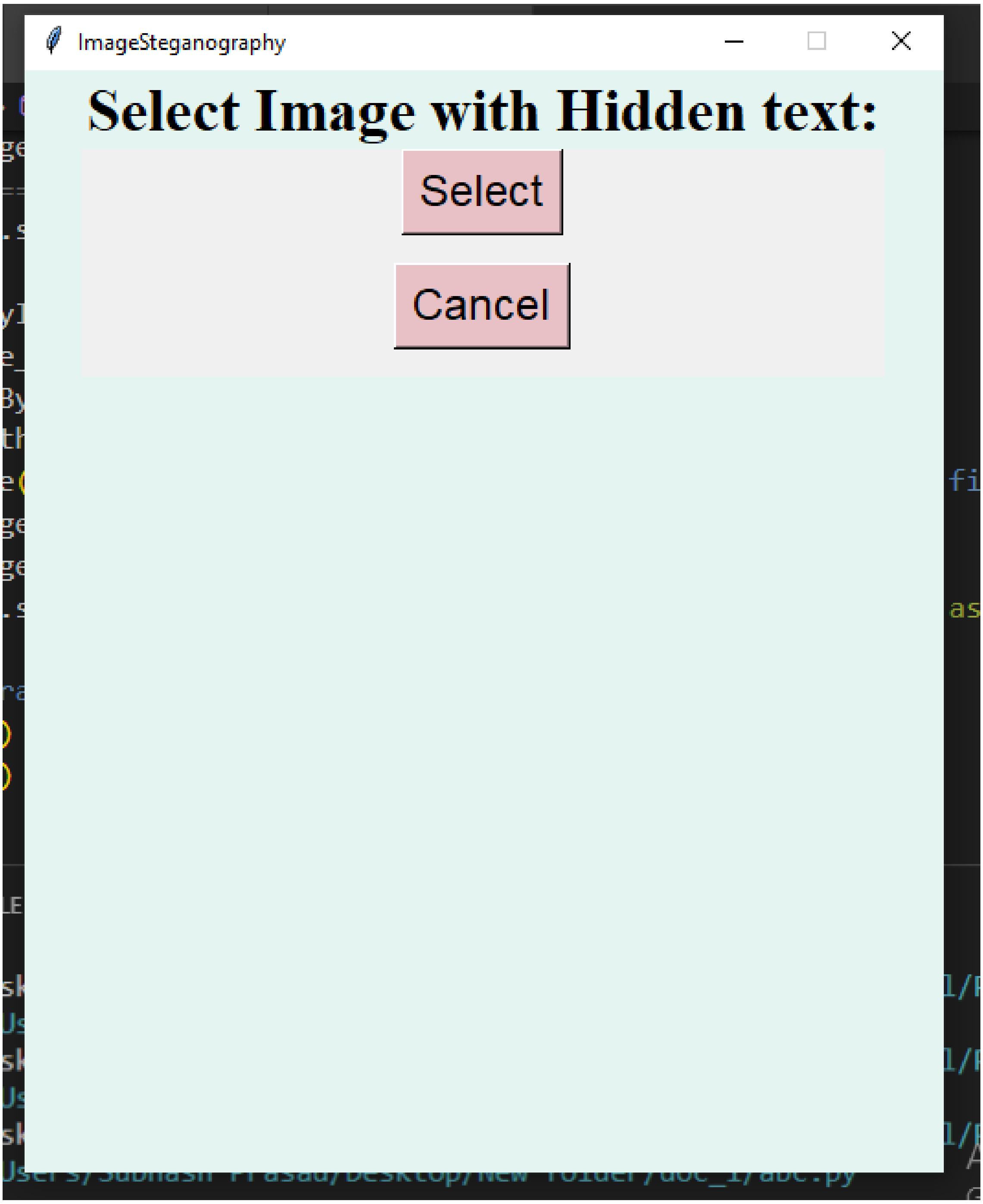


Fig 8.1 Pic 1

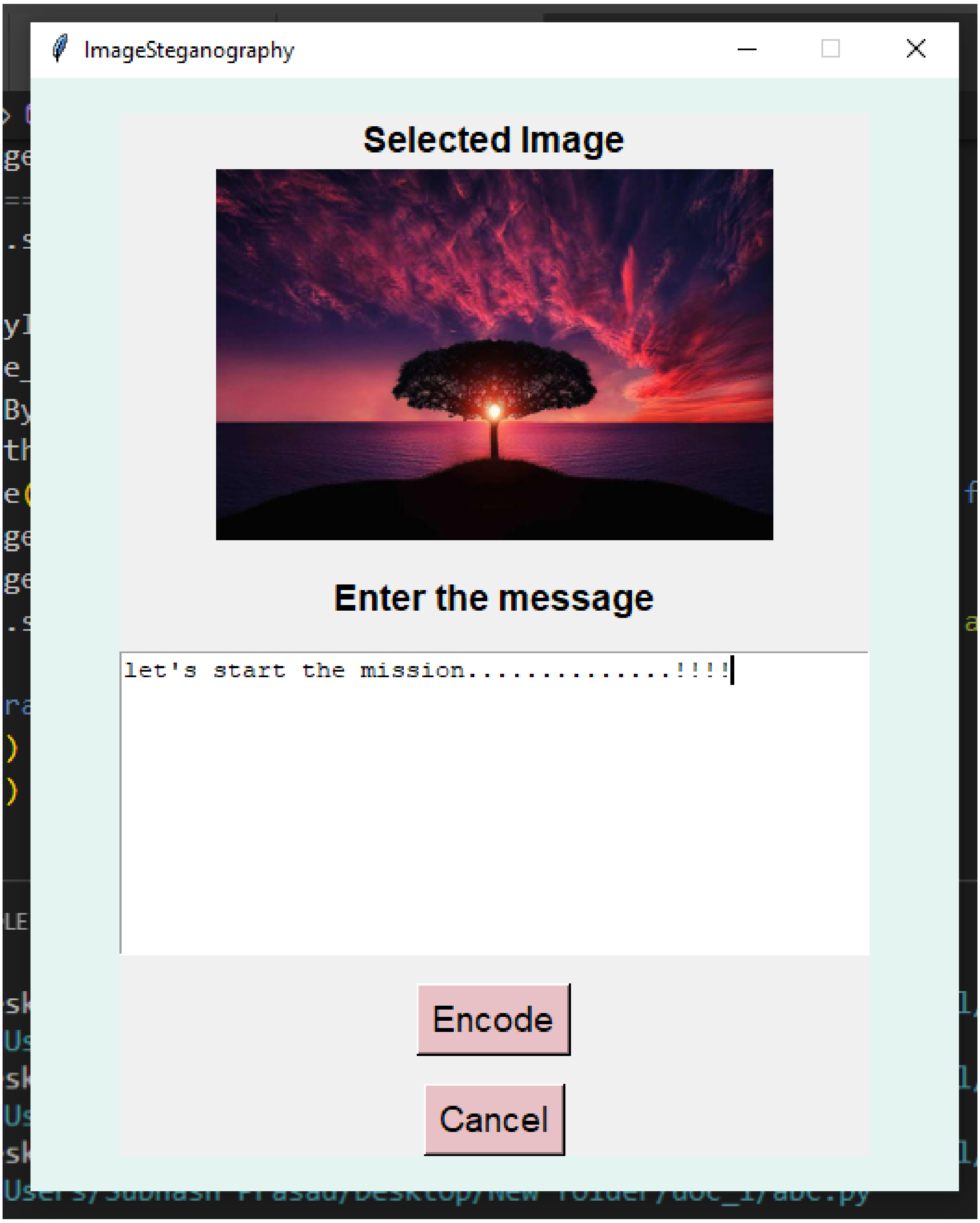
(b) Select any type of file that we have to load.



.png

Fig 8.2 Pic 2

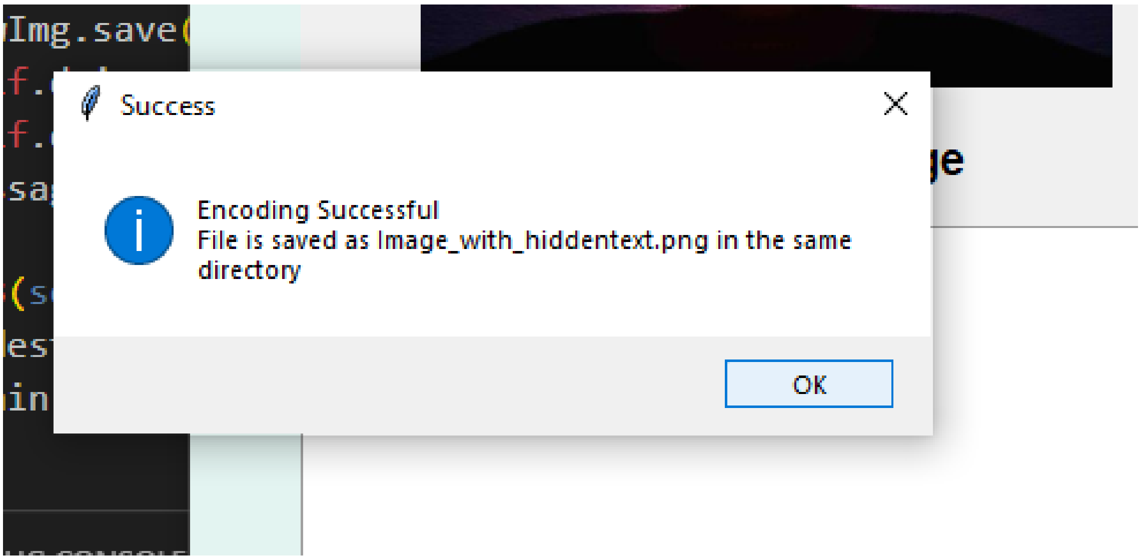
(c) Then we have to write a secret message, which we have to show to receiver end.



let's start the mission...............!!!!

Fig 8.3 Pic 3

(d) Then message is popup that encoding is successfull.



Success

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

Fig 8.4 Pic 4

(e) After that at the receiver end we click on the decode option to see the message.

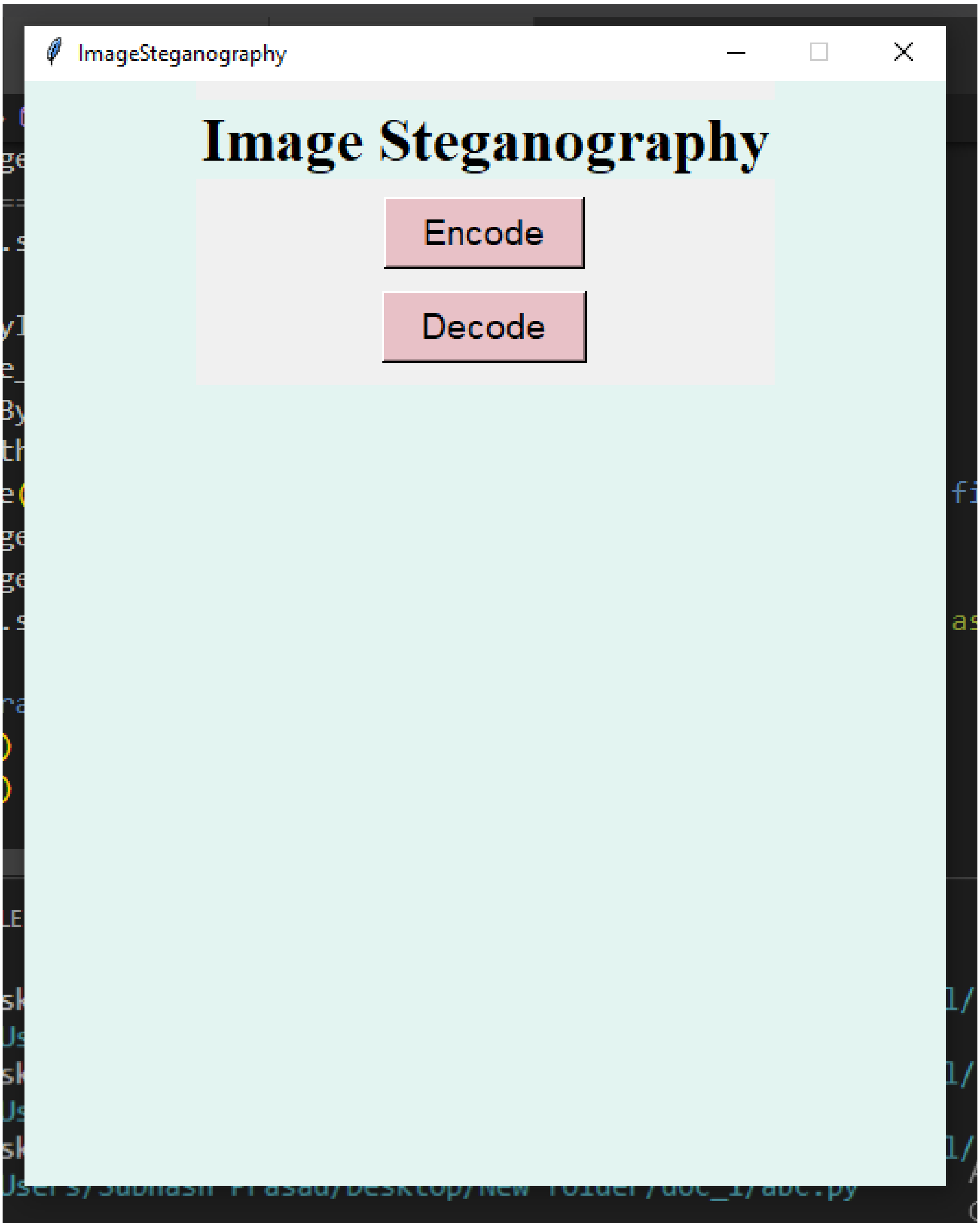
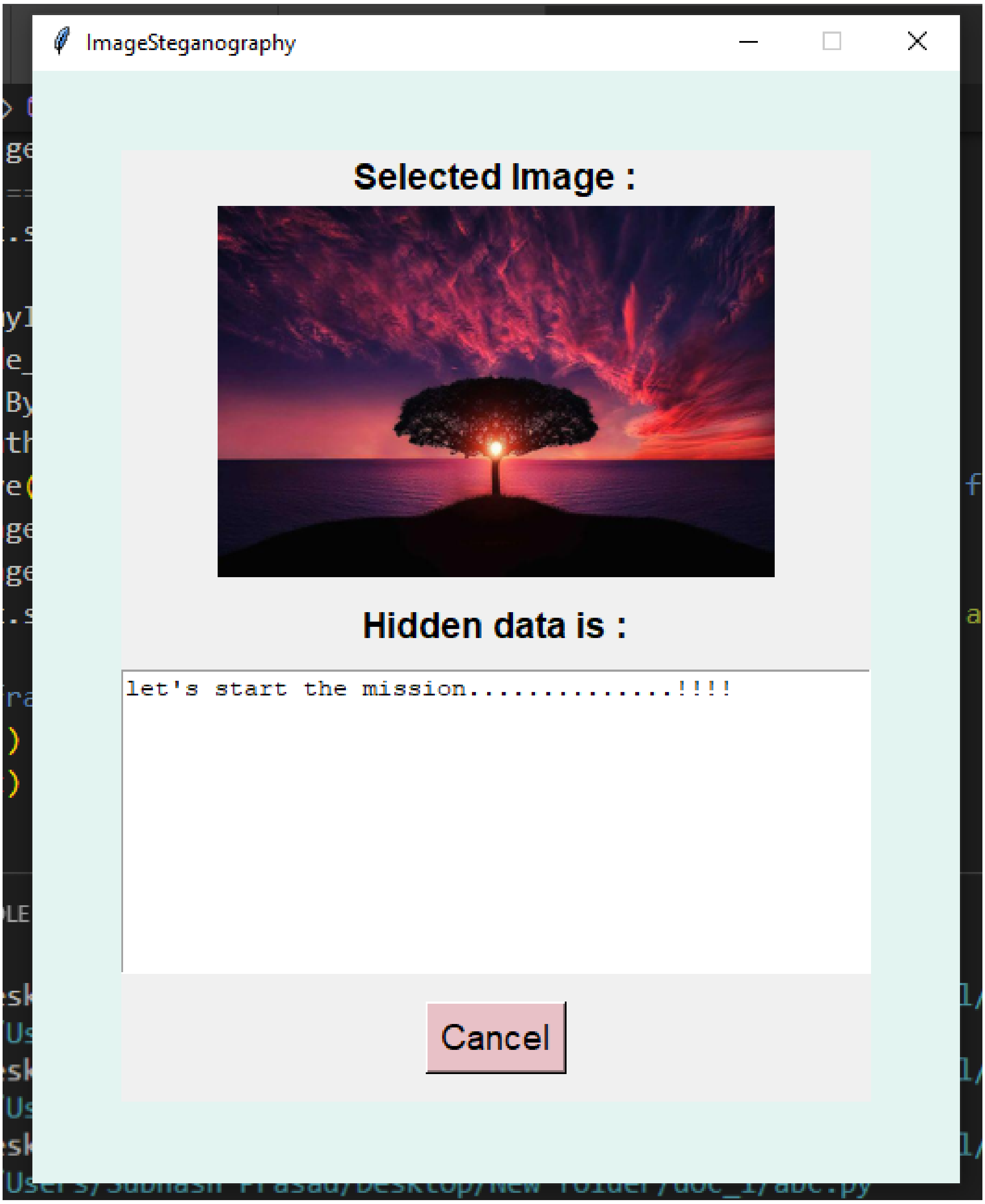


Fig 8.5 Pic 5

(f) Now, the secret message is display through image, After seeing, click on cancel



let's start the mission..............!!!!

Fig 8.6 Pic 6

9 Future Enhancement And Conclusion

Steganography is a really interesting subject and outside of the mainstream cryptography and system administration that most of us deal with day after day. Steganography can be used for hidden communication. We have explored the limits of steganography theory and practice. We printed out the enhancement of the image steganography system using LSB approach to provide a means of secure communication. A stego-key has been applied to the system during embedment of the message into the cover image. This steganography application software provided for the purpose to how to use any type of image formats to hiding any type of files inside their. The master work of this application is in supporting any type of pictures without need to convert to bitmap, and lower limitation on file size to hide, because of using maximum memory space in pictures to hide the file. Since ancient times, man has found a desire in the ability to communicate covertly. The recent explosion of research in watermarking to protect intellectual property is evidence that steganography is not just limited to military or espionage applications. Steganography, like cryptography, will play an increasing role in the future of secure communication in the “digital world”.

References

T.Morkel , J.H.P. Eloff, M.S.Olivier - “An overview of image Steganography” - Information and Computer Security Architecture (ICSA) Research Group Department of Computer Science University of Pretoria, 0002, Pretoria, South Africa.

F. L Bauer, “Decrypted Secrets - Methods and Maxims of Cryptology,” Berlin, Heidelberg, Germany, Springer-Verlag (1997).

Birgit Pfitzmannn, rInformation Hiding Terminologyr, in Proceedings of FirstWork- shop of Information Hiding, Cambridge, U.K. May 30 - June 1, 1996. Lecture Notes in Computer Science, Vol.1174, pp 347-350. Springer-Verlag (1996).

Gustavus J. Simmons, “The Prisonersr Problem and the Subliminal Channel”, in Proceedings of CRYPTO ‘83, pp 51-67. Plenum Press (1984).

Stefan Katzenbeisser and Fabien A. P. Petitcolas, “Information Hiding Techniques for Steganography and Digital Watermarking,” Artech House (2000).

Neil F. Johnson and Sushil Jajodia, “Steganalysis of Images Created using Current Steganography Software/’ in Proceedings of 2nd International Workshop on Information Hiding, April 1998, Portland, Oregon, USA. pp. 273 - 289.

Ross J. Anderson and Fabien A.P. Petitcolas, ron the limits of steganography,r IEEE Journal on Selected Areas in Communications (J-SAC), Special Issue on Copyright Privacy Protection, vol. 16 no. 4, pp 474-481, May 1998.

Scott Craver, “On Public-key Steganography in the Presence of an Active Warden,” in Proceedings of 2nd International Workshop on Information Hiding, April 1998, Portland, Oregon, USA. pp. 355 - 368.

Yeuan-Kuen Lee and Ling-Hwei Chen, “An Adaptive Image Steganographic Model Based on Minimum-Error LSB Replacement,r in Proceedings of the Ninth National Conference on Information Security, pp. 8-15. Taichung, Taiwan, May 14-15, 1999.

Yeuan-Kuen Lee and Ling-Hwei Chen, “A High Capacity Image Steganographic Model,r accepted by lEE Proceedings Vision, Image and Signal Processing. (2000).