# FREEDOM INTERNATIONAL SCHOOL

### # 33, Sector IV, HSR Layout, Bengaluru, Karnataka 560102

SCHOOL CODE: 45175 AFFILIATION NUMBER: 830183

**COMPUTER SCIENCE PROJECT ON**

**CRYPTOGRAPHY**

**SUBMITTED BY**

MANNITH NARAYAN

G.RAM

Class XI A

**Under the guidance of**

**Ms. REJANI.A.NINAN**

**TGT- Computer Science**

**Vice Principal Principal**

**Ms. Clara David Ms. Sneha Rai**

**Freedom International School Freedom International School**

**Bangalore Bangalore**



**FREEDOM INTERNATIONAL SCHOOL**

# 33, Sector IV, HSR Layout, Bengaluru, Karnataka 560102

SCHOOL CODE: 45175 AFFILIATION NUMBER: 830183

**CERTIFICATE**

This is to certify that the Computer Project Report entitled

**Cryptography**

was carried out by MANNITH NARAYAN

of Class **XI A,**

### Roll No. 18 ,

a student of FREEDOM INTERNATIONAL SCHOOL

in partial fulfilment of the Computer Practical Examinations prescribed by the CBSE during the Academic Year 2019-2020.

I certify that this project has been done by him/her with his/her own effort under the guidance of the teacher.

#### Signature of the Teacher in Charge Signature of the Principal

**Ms. Rejani.A. Ninan Ms. Sneha Rai**

**Name of the Examiners Signature with date**

**1.**

**2.**



**FREEDOM INTERNATIONAL SCHOOL**

# 33, Sector IV, HSR Layout, Bengaluru, Karnataka 560102

SCHOOL CODE: 45175 AFFILIATION NUMBER:830183

**CERTIFICATE**

This is to certify that the Computer Project Report entitled

**Cryptography**

was carried out by G.RAM

of Class **XI A,**

### Roll No. 16 ,

a student of FREEDOM INTERNATIONAL SCHOOL

in partial fulfilment of the Computer Practical Examinations prescribed by the CBSE during the Academic Year 2019-2020.

I certify that this project has been done by him/her with his/her own effort under the guidance of the teacher.

#### Signature of the Teacher in Charge Signature of the Principal

**Ms. Rejani.A. Ninan Ms. Sneha Rai**

**Name of the Examiners Signature with date**

**1.**

**2.**

**ACKNOWLEDGEMENT**

I would like to express my special thanks and gratitude to my teacher and project guide (Ms. Rejani.A.Ninan) who gave me this wonderful opportunity to work on this interesting project, which required a lot of research and was an excellent learning experience.

My sincere thanks goes to Ms. Sneha Rai, our Principal, for her coordination in extending every possible support for the completion of this project.

I also thank my parents for their constant motivation and support.

Last but not least, I would like to thank all those who had helped me directly or indirectly towards the completion of this project.

**INDEX**

|  |  |  |
| --- | --- | --- |
| **Sl. No.** | **CONTENT** | **PAGE NO.** |
| 1 | Overview | 1 |
| 2 | Project Description | 2 |
| 3 | Flowchart | 3 |
| 4 | Source Code | 5 |
| 5 | Output | 13 |
| 6 | Further Improvements | 18 |
| 7 | References | 19 |

**OVERVIEW**

Python is a high-level, interpreted, interactive and object-oriented scripting language. Python is designed to be highly readable. It uses English keywords frequently where as other languages use punctuation, and it has fewer syntactical constructions than other languages.

Python was developed by Guido van Rossum in the late eighties and early nineties at the National Research Institute for Mathematics and Computer Science in the Netherlands.

Python is derived from many other languages, including ABC, Modula-3, C, C++, Algol-68, SmallTalk, and UNIX shell and other scripting languages.

**PROJECT DESCRIPTION**

DocLocker is on the principle of encrypting and decrypting the data in a specific file. It helps in ensuring some basic security for the data in the file. The data in the file is presented in a vague form so that it is not understood unless the data is decrypted.

**Software Used**

* Python(IDLE)

1. Tkinter module
2. os module
3. Sys module
4. Cryptography module

**FLOWCHART**

**The user enters the login credentials**

**The login credentials are validated**

**A window appears with text**

**“Invalid Login”**

**A window appears with 2 tabs (File & Profile) with different buttons**

**Valid**

**Invalid**

**The user has to select the tab**

**Profile**

**File**

**A sign out button is displayed**

**An upload button is displayed**

**B**

**A**

**C**

**C**

**A**

**B**

**Select ‘Upload’ to upload a file**

**A key is shown and the encrypted data is shown in the form of a code**

**A decrypt button is shown in a window and the encrypted data is decrypted and shown to the user if clicked**

**SOURCE CODE**

import tkinter as tk

from tkinter import ttk

from tkinter import \*

from tkinter.filedialog import askopenfilename

from tkinter import messagebox

import sys

from cryptography.fernet import Fernet

import os

creds = 'tempfile.temp' # This just sets the variable creds to 'tempfile.temp'

def Signup(): # This is the signup definition,

global pwordE

global nameE

global roots

roots = Tk() # This creates the window

roots.title('Signup')

intruction = Label(roots, text='Please Enter new Credentials\n')

intruction.grid(row=0, column=0, sticky=E)

nameL = Label(roots, text='New Username: ')

pwordL = Label(roots, text='New Password: ')

nameL.grid(row=1, column=0, sticky=W)

pwordL.grid(row=2, column=0, sticky=W)

nameE = Entry(roots)

pwordE = Entry(roots, show='\*')

nameE.grid(row=1, column=1)

pwordE.grid(row=2, column=1)

signupButton = Button(roots, text='Signup', command=FSSignup)

signupButton.grid(columnspan=2, sticky=W)

roots.mainloop()

def browsefunc():

global my\_file

filename=askopenfilename(title='Select A File',filetypes=(('text files','\*.txt'),('all files','\*')))

path=filename

key=Fernet.generate\_key()

print('Key= ',key)

file=open('c:\doclocker\key.key','wb')

file.write(key)#the key will be in type bytes

file.close()

#encrypting the file

input\_file=path

with open(input\_file,'rb') as f:

data=f.read()

f=Fernet(key)

encrypted=f.encrypt(data)

print('encrypted data=',encrypted)

def decrypt():

global my\_file

output\_file='c:\doclocker\encryptedfile.txt'

with open(output\_file,'wb') as f:

f.write(encrypted)

with open(output\_file,'rb') as f:

data=f.read()

print('encrypted content of the file is:', data.decode())

e=Fernet(key)

decrypted\_file=e.decrypt(data)

print('the decrypted content of the file is:',decrypted\_file.decode())

#asking the user whether he has to decrypt the program or just signout

btn=ttk.Button(text='Decrypt', command=decrypt)

btn.pack()

pathlabel=Label()

pathlabel.pack()

def callback():

messagebox.showinfo("About DocLocker","This Application helps you to encrypt each of your uploaded document with a unique password of your choice 'Created by Mannith and Ram'")

def FSSignup():

with open(creds, 'w') as f: # Creates a document using the variable

f.write(nameE.get()) # nameE is the variable we were storing the input

f.write('\n') # Splits the line into username and password

f.write(pwordE.get())

f.close()

roots.destroy()

Login() #this will open the login window

def Login():

global nameEL

global pwordEL

global rootA

rootA = Tk()

rootA.title('Login')

intruction = Label(rootA, text='Please Login\n')

intruction.grid(sticky=E)

nameL = Label(rootA, text='Username: ')

pwordL = Label(rootA, text='Password: ')

nameL.grid(row=1, sticky=W)

pwordL.grid(row=2, sticky=W)

nameEL = Entry(rootA) #input the name

pwordEL = Entry(rootA, show='\*')

nameEL.grid(row=1, column=1)

pwordEL.grid(row=2, column=1)

loginB = Button(rootA, text='Login', command=CheckLogin)

loginB.grid(columnspan=2, sticky=W)

rmuser = Button(rootA, text='Delete User', fg='red', command=DelUser)

rmuser.grid(columnspan=2, sticky=W)

rootA.mainloop()

def CheckLogin():

with open(creds) as f:

data = f.readlines() #the data in the document is in the form of variables

uname = data[0].rstrip()

pword = data[1].rstrip()

if nameEL.get() == uname and pwordEL.get() == pword:

r = Tk() # Opens new window

r.title('Document Locker')

r.geometry('500x500') # Makes the window a certain size

rlbl = Label(r, text='\nwelcome to document locker') # "logged in" label

rlbl.pack() # Pack is like .grid(), just different

rlbl2=Label(r,text='Everyone likes PRIVACY',font='Times 30 bold')

rlbl2.pack()

tabControl=ttk.Notebook(r)#created tabcontrol

tab1=ttk.Frame(tabControl)

tabControl.add(tab1, text='File')#creates a tab named file

tabControl.pack()

tab2=ttk.Frame(tabControl)

tabControl.add(tab2, text='Profile')

tabControl.pack(expand=1, fill='both')

btn1t1=ttk.Button(tab1,text='Upload', command=browsefunc)

btn1t1.pack()

pathlabel=Label(r)

pathlabel.pack()

btn3t1=ttk.Button(tab1, text='Want to know about Us?',command=callback)

btn3t1.pack()

btn2t2=ttk.Button(tab2, text='Signout', command=r.destroy)

btn2t2.pack()

r.mainloop()

else:

r = Tk()

r.title('Document Locker')

r.geometry('150x50')

rlbl = Label(r, text='\n[!] Invalid Login')

rlbl.pack()

r.mainloop()

def DelUser():

os.remove(creds) # Removes the file

rootA.destroy()

Signup()

if os.path.isfile(creds):

Login()

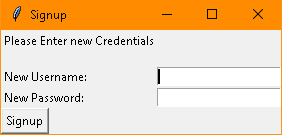
else:

Signup()

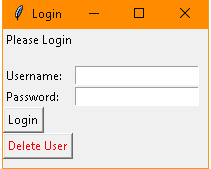
***P.S****: a folder named ‘doclocker’ is to be created in the C drive of the computer. Also, a text filed named as ‘encryptedfile’ has to be created inside the folder named doclocker. The folder stores the key for the encrypted file and the encrypted data is copied into the file ‘encryptedfile’ temporarily. The program, as of now only accepts text documents as an input.*

**OUTPUT**

1. A signup window appears



1. Once the signup is complete. A login window appears



The button ‘Delete User’ deletes the username and password from the tempfile (temporary file) and a signup window is displayed.

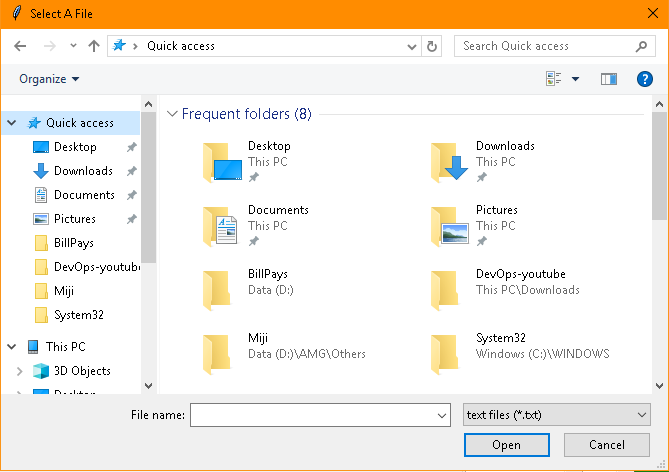
3. Once the login credentials are entered, the main window appears with 2 Tabs



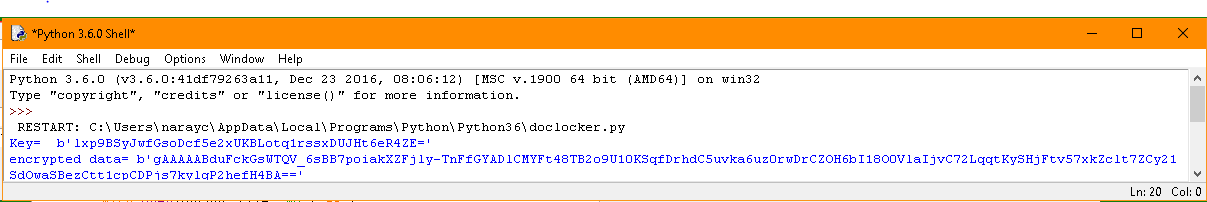
4. If the login is not successful, the following window appears.



5. In order to encrypt a text file, click on the upload button. A dialog box appears

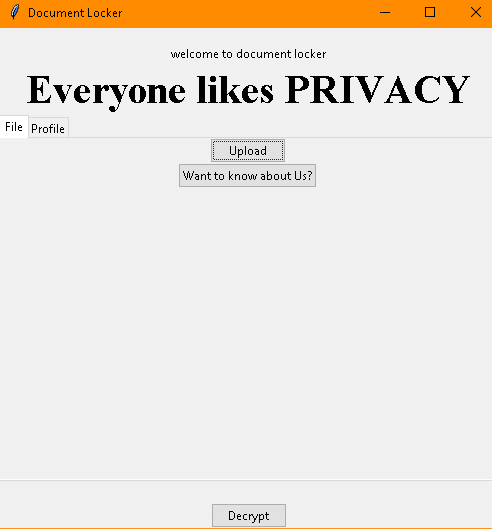


Once a text file is selected, a window with similar information appears (in the python program).

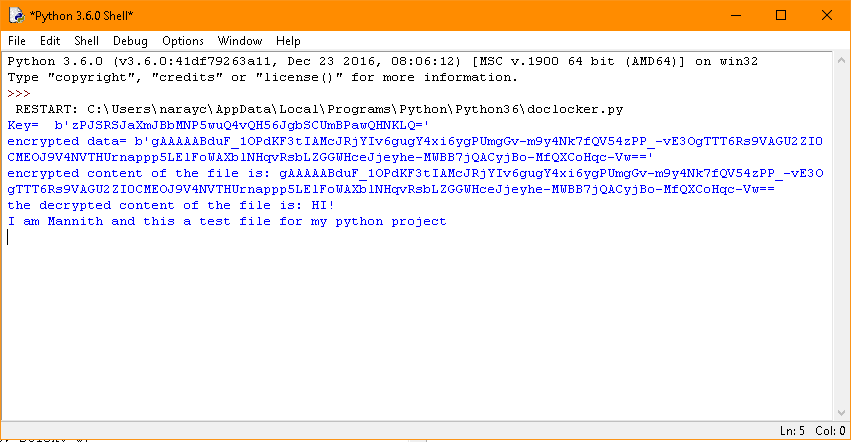


The above image consists of 2 elements ‘key’ and ‘encrypted data’. The data in the selected file is shown to the user in an encrypted form i.e., in the lines of ‘encrypted data’.

The main window now has a button added at the bottom which when clicked decrypts the data

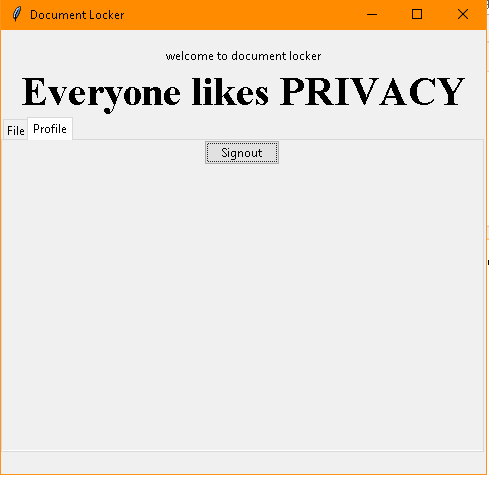


6. Once the Decrypt button is clicked, the python program window shows the decrypted data



1. Once the Decrypt button is clicked, the python program window shows the decrypted data

7.The user can sign-out by clicking on the ‘sign out’ button in the profile tab



**FURTHER IMPROVEMENTS**

* A glitch during the start of the program (opening of an empty window) has to be solved.
* The decrypted data can be sent to the user’s desired email address rather than showing it to the user at the same moment.
* A better looking GUI

REFERENCES

1. Sourcecodester
2. YouTube- PyTutorials

-TheCodex

-Edureka!