**University of Prince Mugrin**

**College of Computer and Cyber Sciences**

**Computer Science Department**

**CS111**

**CRYPTOGRAPHY**

**Student Name: Raghed Alrefi || L01**

**Student ID: 4112248**

**Student Name: Tasnim Mohammad || L06**

**Student ID: 4110400**

**Student Name: Khadija Mohammad Khan || L06**

**Student ID: 4111576**

**Student Name: Lama Mohammad || L06**

**Student ID: 4210682**

\_\_\_\_\_\_

**University Supervisor(S):**

**Basimah Aljhne**

**College of Computer and Cyber Sciences**

**Intellectual Property Right Declaration**

This is to declare that the work under the supervision of \_\_Basimah Aljhne \_\_\_\_ having title “ secret meassage (cryptography)\_\_\_\_\_\_” carried out in partial fulfillment of the requirements of the course \_\_\_\_\_\_CS111\_\_\_\_\_, is the sole property of the University of Prince Mugrin and the respective supervisor and is protected under the intellectual property right laws and conventions. It can only be considered/ used for purposes like extension for further enhancement, product development, adoption for commercial/organizational usage, etc., with the permission of the University and respective supervisor.

**Anti-Plagiarism Declaration**

This is to declare that the above publication produced under the supervision of\_\_\_\_\_ Basimah Aljhne \_\_\_\_\_\_\_\_\_having title “\_\_\_\_\_\_\_\_\_secret meassage (cryptography)\_\_\_\_\_\_” is the sole contribution of the author(s) and no part hereof has been reproduced illegally (cut and paste) which can be considered as Plagiarism. All referenced parts have been used to argue the idea and have been cited properly. I/We will be responsible and liable for any consequence if violation of this declaration is p

This above statement applies to all students and faculty members

Table of Contents

[Introduction 3](#_Toc90512047)

[Project Idea 3](#_Toc90512048)

[Algorithm (Flow Chart): 4](#_Toc90512049)

[5](#_Toc90512050)

[Features of Design and implementation that worked well: 6](#_Toc90512051)

[1. Feature 1 6](#_Toc90512052)

[2. Feature 2 6](#_Toc90512053)

[3. Feature 3 6](#_Toc90512054)

[4. Feature 4 6](#_Toc90512055)

[Features of Design and implementation that did not worked well: 10](#_Toc90512056)

[Ideas for additional features and design improvements 11](#_Toc90512057)

[1. Feature 1 11](#_Toc90512058)

[2. Featu re 2 11](#_Toc90512059)

[Reference 12](#_Toc90512061)

[Appendi ces 13](#_Toc90512062)

[***The Source Code***](#_Appendices)**……..…………………………………………………….……….…..…………10**

[***Work distributes***](#_Appendices)**……………………………..………….…………………………….….………24**

# Introduction

Privacy is a hard thing to maintain in our modern technology days. And cryptography is one way to solve this problem.

We use it to safeguard personal identifiable information (PII) and other confidential data.

# Project Idea

our project is a program called **CRYPTOGRAPHY**. The program will make the users personal life private by encrypting their private messages without the knowledge of other people. You can communicate with your secret person without the knowledge of other people, and that's what our high privacy program provides. It is a completely error-free program.

A picture containing icon

Description automatically generated

Note:

To run the program with your personal device you must do the following:

* Install Montserrat font
* Save the special image of the program, also add the location of the image in the code like its shown “C:\\Users\\YourPCNmae\\LocationFolder\\FolderName\\ImageName.png”
* Install pyperclip module "pip install pyperclip"

All the requirements you can find them here (https://drive.google.com/drive/folders/1JIcWTy7zLynNtRcT7By8FXghW77HoNZl?usp=sharing)

# Algorithm (Flow Chart):

**ASCII Code Encrypt ASCII code Decrypt**

Diagram

Description automatically generatedDiagram

Description automatically generated

**Caesar Cipher Encrypt Caesar Cipher Decrypt**

Diagram

Description automatically generatedDiagram

Description automatically generated

Diagram

Description automatically generatedDiagram

Description automatically generated **Morse Code Encrypt Morse Code Decrypt**

Diagram

Description automatically generated

**Baconian Cipher Encrypt Baconian Cipher Decrypt**

Diagram

Description automatically generated

# Features of Design and implementation that worked well:

## Feature 1

The rest buttons

It will clear the entry box and the list box

## Feature 2

Saving the encrypted and decrypted message in file. Also, the user will choose the name of the file and the location

And it will automatically open the file after saving it

## Feature 3

The encrypted message will be copied auto in clipboard

## Feature 4

To make the message more personal and private between the sender and the receiver we provide a key (shift) in the Creaser cipher algorithm . it is only known between the sender and the receiver so that no one can discover what in the message

A screenshot of a computer

Description automatically generated with medium confidence

Graphical user interface, application, Teams

Description automatically generatedGraphical user interface, application, Teams

Description automatically generated

Graphical user interface, application

Description automatically generatedA picture containing text, monitor, screen, indoor

Description automatically generated

A picture containing text, monitor, screen, indoor

Description automatically generatedA picture containing text, monitor, indoor, screenshot

Description automatically generated

# Features of Design and implementation that did not worked well:

We faced a problem in the fourth algorithm, Baconian cipher decrypt didn’t work If the user enters any character except letters from a-z. That’s why we used try and except to inform the user with this condition.

Anything else in the project worked well.

**ERROR-FREE Program**

# Ideas for additional features and design improvements

## Feature 1

We'll allow the user to add an encryption method with ease

## Feature 2

## One of the future additions is that the encryption message will appear only by solving a puzzle sent by the sender and only the user knows how to solve it

## Feature 3

We want to include Arabic Language too, instead of just english

# Reference

Basss. (2020, March 17). *“how to reset entry in tkinter” Code Answer’s*. Retrieved from codegrepper: https://www.codegrepper.com/code examples/python/how+to+reset+entry+in+tkinter

*How can I open files in external programs in Python?* . (2013, February 24). Retrieved from stackoverflow: https://stackoverflow.com/questions/15054434/how-can-i-open-files-in-external-programs-in-python

Nigam, P. (2021, Novamber 23). *Morse Code Translator In Python*. Retrieved from geeksforgeeks.org: https://www.geeksforgeeks.org/morse-code-translator-python/

*Quick and easy file dialog in Python?* (2012, February 16). Retrieved from stackoverflow: https://stackoverflow.com/questions/9319317/quick-and-easy-file-dialog-in-python

*Welcome to Pyperclip’s documentation!* (n.d.). Retrieved from github: https://github.com/asweigart/pyperclip/blob/master/docs/index.rst

# Appendices

**Appendix A The source code**

from tkinter import \*

from tkinter import messagebox,filedialog,simpledialog

import pyperclip #module for copy to clipboard

import os

#----------------------------------------------------------------------------------------------------

def Caeser\_Cipher\_encrypt():

    """function to encrypt a message based on ceaser cipher algorithm"""

    message= e1.get()

    message=message.lower()

    alphabet = "abcdefghijklmnopqrstuvwxyz"

    encrypted\_messege = ""

    key= simpledialog.askinteger("Key","Please Enter a key")

    for ch in message:

        if ch in alphabet:

            index = alphabet.find(ch)

            newindex = (index + key) % 26

            newCharacter = alphabet[newindex]

            encrypted\_messege += newCharacter

        else:

            encrypted\_messege += ch

    messagebox.showinfo("Result",("Your super secret message is ",encrypted\_messege))

    pyperclip.copy(encrypted\_messege) #to copy the message to the clipboard

#----------------------------------------------------------------------------------------------------

def Caeser\_Cipher\_decrypt():

    """function to decrypt a message based on ceaser cipher algorithm"""

    message= e1.get()

    message=message.lower()

    alphabet = "abcdefghijklmnopqrstuvwxyz"

    decrypted\_messege = ""

    key= simpledialog.askinteger("Key","Please Enter a key")

    for ch in message:

        if ch in alphabet:

            index = alphabet.find(ch)

            newindex = (index - key) % 26

            newCharacter = alphabet[newindex]

            decrypted\_messege += newCharacter

        else:

            decrypted\_messege += ch

    messagebox.showinfo("Result",("Your super secret message is ",decrypted\_messege))

    pyperclip.copy(decrypted\_messege) #to copy the message to the clipboard

#----------------------------------------------------------------------------------------------------

def ASCII\_encrypt():

    """function to enecrypt the message based on ASCII code algorithm"""

    message= e1.get()

    result=""

    for i in message:

        encrypted\_letter=0

        o=ord(i)

        encrypted\_letter=o+10

        encrypted\_letter=chr(encrypted\_letter)

        result+=encrypted\_letter

    messagebox.showinfo("Result",("Your super secret message is ",result))

    pyperclip.copy(result)

#----------------------------------------------------------------------------------------------------

def ASCII\_decrypt():

    """function to decrypt the message based on ASCII code algorithm"""

    message= e1.get()

    result=""

    for i in message:

        decrypted\_letter=0

        do=ord(i)

        decrypted\_letter=do-10

        decrypted\_letter=chr(decrypted\_letter)

        result+=decrypted\_letter

    messagebox.showinfo("Result",("Your decrypt message is ",result))

    pyperclip.copy(result)

#----------------------------------------------------------------------------------------------------

def Morse\_code\_encrypt():

    """function to encrypt a message based on morse code cipher algorithm"""

    # resourse is (https://www.geeksforgeeks.org/morse-code-translator-python/)

    message= e1.get()

    message=message.upper()

    encrypted\_messege = ""

    MORSE\_CODE\_DICT = { 'A':'.-', 'B':'-...',

                    'C':'-.-.', 'D':'-..', 'E':'.',

                    'F':'..-.', 'G':'--.', 'H':'....',

                    'I':'..', 'J':'.---', 'K':'-.-',

                    'L':'.-..', 'M':'--', 'N':'-.',

                    'O':'---', 'P':'.--.', 'Q':'--.-',

                    'R':'.-.', 'S':'...', 'T':'-',

                    'U':'..-', 'V':'...-', 'W':'.--',

                    'X':'-..-', 'Y':'-.--', 'Z':'--..',

                    '1':'.----', '2':'..---', '3':'...--',

                    '4':'....-', '5':'.....', '6':'-....',

                    '7':'--...', '8':'---..', '9':'----.',

                    '0':'-----', ', ':'--..--', '.':'.-.-.-',

                    '?':'..--..', '/':'-..-.', '-':'-....-',

                    '(':'-.--.', ')':'-.--.-'}

    for letter in message:

        if letter != ' ':

            encrypted\_messege += MORSE\_CODE\_DICT[letter] + ' ' #space to serapate diffrent characters

        else:

            # 1 space indicates different characters

            # and 2 indicates different words

            encrypted\_messege += ' '

    messagebox.showinfo("Result",("Your super secret message is ",encrypted\_messege))

    pyperclip.copy(encrypted\_messege) #to copy the message to the clipboard

#----------------------------------------------------------------------------------------------------

def Morse\_code\_decrypt():

    """function to encrypt a message based on morse code cipher algorithm"""

    # resourse is (https://www.geeksforgeeks.org/morse-code-translator-python/)

    message= e1.get()

    message += ' '

    decrypted\_messege = ""

    citext = ""

    MORSE\_CODE\_DICT = { 'A':'.-', 'B':'-...',

                    'C':'-.-.', 'D':'-..', 'E':'.',

                    'F':'..-.', 'G':'--.', 'H':'....',

                    'I':'..', 'J':'.---', 'K':'-.-',

                    'L':'.-..', 'M':'--', 'N':'-.',

                    'O':'---', 'P':'.--.', 'Q':'--.-',

                    'R':'.-.', 'S':'...', 'T':'-',

                    'U':'..-', 'V':'...-', 'W':'.--',

                    'X':'-..-', 'Y':'-.--', 'Z':'--..',

                    '1':'.----', '2':'..---', '3':'...--',

                    '4':'....-', '5':'.....', '6':'-....',

                    '7':'--...', '8':'---..', '9':'----.',

                    '0':'-----', ', ':'--..--', '.':'.-.-.-',

                    '?':'..--..', '/':'-..-.', '-':'-....-',

                    '(':'-.--.', ')':'-.--.-'}

    for letter in message:

        if (letter != ' '):

            i = 0

            citext += letter

        else:

            i += 1

            if i == 2 :

                decrypted\_messege += ' '

            else:

                decrypted\_messege += list(MORSE\_CODE\_DICT.keys())[list(MORSE\_CODE\_DICT.values()).index(citext)]

                citext = ''

    decrypted\_messege=decrypted\_messege.lower()

    messagebox.showinfo("Result",("Your super secret message is ",decrypted\_messege))

    pyperclip.copy(decrypted\_messege) #to copy the message to the clipboard

#----------------------------------------------------------------------------------------------------

def Baconian\_cipher\_encrypt():

    """function to encrypt a message based on Baconian cipher algorithm"""

    baconian={'A':'aaaaa ', 'B':'aaaab ', 'C':'aaaba ', 'D':'aaabb ','E':'aabaa ',

    'F':'aabab ', 'G':'aabba ', 'H':'aabbb ', 'I':'abaaa ', 'J':'abaab ',

    'K':'abaab ', 'L':'ababa ', 'M':'ababb ', 'N':'abbaa ', 'O':'abbab ',

    'P':'abbba ', 'Q':'abbbb ', 'R':'baaaa ', 'S':'baaab ', 'T':'baaba ',

    'U':'babaa ', 'V':'babab ', 'W':'babaa ', 'X':'babab ', 'Y':'babba ',

    'Z':'babbb '}

    message= e1.get()

    message=message.upper()

    encrypted\_messege = ""

    for ch in message:

        if(ch != ' '):

            encrypted\_messege += baconian[ch]

        else:

            # adds space

            encrypted\_messege += ' '

    messagebox.showinfo("Result",("Your super secret message is ",encrypted\_messege))

    pyperclip.copy(encrypted\_messege)

#----------------------------------------------------------------------------------------------------

def Baconian\_cipher\_decrypt():

    """function to decrypt a message based on Baconian cipher algorithm"""

    baconian={'aaaaa ':'A', 'aaaab ':'B', 'aaaba ':'C', 'aaabb ':'D','aabaa ':'E',

    'aabab ':'F', 'aabba ':'G', 'aabbb ':'H', 'abaaa ':'I', 'abaab ':'J',

    'abaab ':'K', 'ababa ':'L', 'ababb ':'M', 'abbaa ':'N', 'abbab ':'O',

    'abbba ':'P', 'abbbb ':'Q', 'baaaa ':'R', 'baaab ':'S', 'baaba ':'T',

    'babaa ':'U', 'babab ':'V', 'babaa ':'W', 'babab ':'X', 'babba ':'Y',

    'babbb ':'Z'}

    message= e1.get()

    decrypted\_messege = ""

    try:

        for i in range (0,len(message),6):

            decrypted\_messege=decrypted\_messege+baconian[message[i:i+6]]

        decrypted\_messege=decrypted\_messege.lower()

        messagebox.showinfo("Result",("Your super secret message is ",decrypted\_messege))

        pyperclip.copy(decrypted\_messege)

    except Exception:

        messagebox.showerror("Error",("Please Try again, this algorithm take only letter from a-z without a space "))

#----------------------------------------------------------------------------------------------------

#the main encryption funtion

def clicked\_encrypt():

    try:

        message= e1.get()

        algorithm=list1.get(list1.curselection()[0])

        if message != "" :  #to make sure that the user enterd a massage

            if algorithm == "Caeser Cipher":

                Caeser\_Cipher\_encrypt()

            elif algorithm == "Morse Code Cipher":

                Morse\_code\_encrypt()

            elif algorithm == "Baconian Cipher":

                Baconian\_cipher\_encrypt()

            elif algorithm == "ASCII Code Cipher":

                ASCII\_encrypt()

        else:

           messagebox.showwarning("Warning",("Please enter a massage"))

    except IndexError:

        #this error occur when the user does not select an algorithm from the listbox

        messagebox.showwarning("Warning",("Please choose an algorithem for encryption  "))

#-----------------------------------------------------------------------------------------

#the main decryption funtion

def clicked\_decrypt():

    try:

        message= e1.get()

        algorithm=list1.get(list1.curselection()[0])

        if message != "" : #to make sure that the user enterd a massage

            if algorithm == "Caeser Cipher":

                Caeser\_Cipher\_decrypt()

            elif algorithm == "Morse Code Cipher":

                Morse\_code\_decrypt()

            elif algorithm == "Baconian Cipher":

                Baconian\_cipher\_decrypt()

            elif algorithm == "ASCII Code Cipher":

                ASCII\_decrypt()

        else:

           messagebox.showwarning("Warning",("Please enter a massage"))

    except IndexError:

        messagebox.showwarning("Warning",("Please choose an algorithem for decryption "))

#----------------------------------------------------------------------------------------------------

def clicked\_reset():

    """to clear message entry and listbox"""

    #refrence is (https://www.codegrepper.com/code-examples/python/how+to+reset+entry+in+tkinter)

    e1.delete(0, END)

    list1.selection\_clear(list1.curselection())

#----------------------------------------------------------------------------------------------------

def save\_file():

    #refrense is :(https://stackoverflow.com/questions/9319317/quick-and-easy-file-dialog-in-python)

    message= e1.get()

    file\_path = filedialog.asksaveasfilename() #make the user choose the location and file name

    f= open(file\_path+".txt","w+")

    new\_messege=pyperclip.paste() # to paste the clipboard to the file

    f.write(("Your Message is: "+message+"\nYour New Message is: "+new\_messege))

    f.close()

    #refresne is (https://stackoverflow.com/questions/15054434/how-can-i-open-files-in-external-programs-in-python)

    fileName = file\_path+".txt"

    os.system("notepad.exe " + fileName) #to automaticly open the file

#----------------------------------------------------------------------------------------------------

#GUI

root = Tk()

root.title("CRYPTOGRAPHY")

root.geometry("841x598")

root.configure(bg = "#131D34")

root.resizable(width=False,height=False) #to make the root window as the same size and can not be bigger

#adding the left background

frame1 = Frame(

    root,

    bg = "#192542",

    height = 598,

    width = 370,

    highlightthickness = 0,

    relief = "ridge")

frame1.place(x = 0, y = 0)

#adding center background

frame2 = Frame(

    root,

    bg = "#192542",

    height =500,

    width =385,

    bd = 0,

    highlightthickness = 0,

    relief = "ridge")

frame2.place(x = 415, y = 63)

#adding img

img = PhotoImage(file = f"C:\\Users\\Raghed Alrefi\\Desktop\\HI\\logo.png")

panel = Label(

    root,

    bg = "#192542",

    image = img)

panel.place(x =50, y = 90)

#adding labels

l1=Label(

    root,

    fg = "#7E53D6",

    bg = "#192542",

    text="CRYPTOGRAPHY",

    font = ("Montserrat", 25, 'bold'))

l1.place(x = 45, y = 346)

l2=Label(root,

    fg = "#70D8FA",

    bg = "#192542",

    text="Welcome to cryptograpgy, the best  \nprogram to hide your masseges.",

    font = ("Montserrat",11),padx=10, pady=5,justify = 'left')

l2.place(x = 38, y = 392)

l3=Label( root,

    fg = "#70D8FA",

    bg = "#192542",

    text="Enter the Massage",

    font = ("Montserrat",13,'bold'))

l3.place(x=442,y=103)

l4=Label(root,

    fg = "#70D8FA",

    bg = "#192542",

    text="Choose Encryption Algorithms",

    font = ("Montserrat",13,'bold'))

l4.place(x=442,y=236)

#adding entries

e1 = Entry(

    root,

    bd = 0,

    bg = "#202F52",

    font = ("Montserrat",9,),

    fg = "white")

e1.place(x = 445, y = 133, width = 324, height = 84)

e2 = Entry(

    bd = 0,

    bg = "#202F52",

    font = ("Montserrat",9,),

    fg = "white")

e2.place(x = 445, y =266, width = 324, height = 35)

#adding listbox

list1=Listbox(

    root,

    height=4,

    bd = 0,

    bg = "#202F52",

    font = ("Montserrat",11),

    fg = "white",

    highlightthickness = 0,

    selectbackground="#7E53D6")

list1.insert(2,"ASCII Code Cipher")

list1.insert(3,"Baconian Cipher")

list1.insert(0,"Caeser Cipher")

list1.insert(1,"Morse Code Cipher")

list1.place(x = 445, y =266, width = 324, height =85)

#adding buttons

b1 = Button(

    root,

    bg = "#70D8FA",

    bd = 0,

    command = clicked\_encrypt,text="Encrypt",

    font = ("Montserrat",12,'bold'),

    fg = "#1A243E",

    relief = "flat")

b1.place(x =446.13, y = 370, width = 147.26, height = 44.96)

b2 = Button(

    root,

    bg = "#7E53D6",

    bd = 0,

    command = clicked\_decrypt,

    text="Decrypt",

    font = ("Montserrat",12,'bold'),

    fg = "white",

    relief = "flat")

b2.place(x =621.61, y = 370, width = 147.26, height = 44.96)

b3 = Button(

    root,

    bg = "#202F52",

    bd = 0,

    command = clicked\_reset,

    text="Reset",

    font = ("Montserrat",12, 'bold'),

    fg = "white",

    relief = "flat")

b3.place(x =532.81, y = 430, width = 147.26, height = 44.96)

b3 = Button(

    root,

    bg = "#202F52",

    bd = 0,

    command = save\_file,

    text="Save in a file",

    font = ("Montserrat",12, 'bold'),

    fg = "white",

    relief = "flat")

b3.place(x =532.81, y = 490, width = 147.26, height = 44.96)

root.mainloop()

**Appendix B work distribute**

|  |  |
| --- | --- |
| Member Name | Feature |
| *Raghad Alrefi* | **Morse Code – SaveFile Button** |
| *Tasneem Mohammad* | **Caesar Cipher - GUI** |
| *Khadija Mohammad* | **Baconian Cipher – Main Encrypt and Decrypt function** |
| *Lama Mohammed* | **ASCII code cipher– Rest Button** |