

```

# import tkinter module
from tkinter import *

# import other necessary modules
import random

# Vigenère cipher for encryption and decryption
import base64

# creating root object
root = Tk()

# defining size of window
root.geometry("650x400"),

# setting up the title of window
root.title("SECURE MESSAGING")

Tops = Frame(root, width=800, relief=SUNKEN)
Tops.pack(side=TOP)

f1 = Frame(root, width=800, relief=SUNKEN)
f1.pack(side=LEFT)

# =====

lblInfo = Label(Tops, font=('Times New Roman', 20, 'bold'),
                text="SECRET MESSAGING \n BY PAVAN V ",
                fg="BLUE", bd=10, anchor='w')

lblInfo.grid(row=0, column=0)

# Initializing variables
Msg = StringVar()
key = StringVar()
mode = StringVar()
Result = StringVar()

# labels for the message
lblMsg = Label(f1, font=('Times New Roman', 12, 'bold'),
               text="MESSAGE", bd=16, anchor="w")

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

# Entry box for the message
txtMsg = Entry(f1, font=('Times New Roman', 12, 'bold'),

```

```

        textvariable=Msg, bd=10, insertwidth=4,
        bg="dark blue", justify='left')

txtMsg.grid(row=1, column=3)
# labels for the key
lblkey = Label(f1, font=('Times New Roman', 12, 'bold'),
               text="KEY (Only Integer)", bd=16, anchor="w")

lblkey.grid(row=2, column=2)

# Entry box for the key
txtkey = Entry(f1, font=('Times New Roman', 12, 'bold'),
               textvariable=key, bd=10, insertwidth=4,
               bg="dark blue", justify='left')

txtkey.grid(row=2, column=3)

# labels for the mode
lblmode = Label(f1, font=('Times New Roman', 12, 'bold'),
                text="MODE(e for encrypt, d for decrypt)",
                bd=12, anchor="w")

lblmode.grid(row=3, column=2)
# Entry box for the mode
txtmode = Entry(f1, font=('Times New Roman', 12, 'bold'),
                textvariable=mode, bd=10, insertwidth=4,
                bg="dark blue", justify='left')

txtmode.grid(row=3, column=3)

# labels for the result
lblResult = Label(f1, font=('Times New Roman', 12, 'bold'),
                  text="The Result-", bd=12, anchor="w")

lblResult.grid(row=4, column=2)

# Entry box for the result
txtResult = Entry(f1, font=('Times New Roman', 12, 'bold'),
                  textvariable=Result, bd=10, insertwidth=4,
                  bg="dark blue", justify='left')

txtResult.grid(row=4, column=3)

# Vigenère cipher

# Function to encode

```

```

def encode(key, msg):
    enc = []
    for i in range(len(msg)):
        key_c = key[i % len(key)]
        enc_c = chr((ord(msg[i]) +
                     ord(key_c)) % 256)
        enc.append(enc_c)
        print("enc:", enc)
    return base64.urlsafe_b64encode("".join(enc).encode()).decode()

# Function to decode

def decode(key, enc):
    dec = []

    enc = base64.urlsafe_b64decode(enc).decode()
    for i in range(len(enc)):
        key_c = key[i % len(key)]
        dec_c = chr((256 + ord(enc[i]) -
                     ord(key_c)) % 256)

        dec.append(dec_c)
        print("dec:", dec)
    return "".join(dec)

def Results():
    # print("Message= ", (Msg.get()))

    msg = Msg.get()
    k = key.get()
    m = mode.get()

    if (m == 'e'):
        Result.set(encode(k, msg))
    else:
        Result.set(decode(k, msg))

# exit function

def qExit():
    root.destroy()

# Function to reset the window

def Reset():

```

```
Msg.set("")
key.set("")
mode.set("")
Result.set("")

# Show message button
btnTotal = Button(f1, padx=10, pady=8, bd=12, fg="black",
                  font=('Times New Roman', 9, 'bold'),
                  width=7, text="Show Message", bg="green",
                  command=Results).grid(row=9, column=2)

# Reset button
btnReset = Button(f1, padx=10, pady=8, bd=12, fg="black",
                  font=('Times New Roman', 10, 'bold'),
                  width=6, text="Reset", bg="orange",
                  command=Reset).grid(row=9, column=3)

# Exit button
btnExit = Button(f1, padx=10, pady=8, bd=12, fg="black",
                  font=('Times New Roman', 10, 'bold'),
                  width=6, text="Exit", bg="red",
                  command=qExit).grid(row=9, column=4)

# keeps window alive
root.mainloop()
```

In this software messages can be encrypted and decrypted. Steps to encrypt and decrypt in this software are listed below

1. First enter the required message in the respective dialog box
2. Now enter a key code which like a verification code which will only be known for sender and receiver.
3. If it is the sender then use “E” in the dialog box, If it is the receiver then use “D” in the dialog box.
4. Finally, in the result dialog box you will get the respective message (encrypted/Decrypted) after clicking the Show message button
5. If you want to reset all the dialog box’s then click on reset Button.
6. To Exit you can click on Exit button.

The screenshot shows a software window titled "PRIVATE MESSAGING" with standard Windows window controls (minimize, maximize, close). The main content area has a light gray background and contains the following elements:

- Title:** "Encryption and Decryption with Verification BY PAVAN V" displayed in blue, bold, serif font.
- Input Fields:** Four blue rectangular input boxes with black borders and drop shadows, arranged vertically. The labels to their left are:
 - MESSAGE
 - KEY (Only Integer)
 - MODE(e for encrypt, d for decrypt)
 - The Result-
- Buttons:** Three buttons are located at the bottom:
 - A green button labeled "Show Message" on the left.
 - A yellow button labeled "Reset" in the center.
 - A red button labeled "Exit" on the right.