Here's a line-by-line explanation of the code:

Importing Libraries

```
from tkinter import *
from tkinter import messagebox
import base64
import os
```

- from tkinter import *: Imports all components from the tkinter module, which is used for creating graphical user interfaces.
- from tkinter import messagebox: Imports the messagebox module for showing message boxes.
- import base64: Imports the base64 module, which allows for encoding and decoding data in Base64.
- import os: Imports the os module, which provides functions for interacting with the operating system (though it is not used in the provided code).

Decryption Function

```
def decrypt():
   password=code.get()
```

- def decrypt(): Defines the decrypt function, which will handle the decryption process.
- password=code.get(): Retrieves the entered password from the Entry widget associated with the code variable.

python
```
 if password=="1234":
```

- Checks if the entered password is "1234".

python
```
    screen2=Toplevel(screen)
    screen2.title("decryption")
    screen2.geometry("400x200")
    screen2.configure(bg="#00bd56")
```

- Creates a new window screen2 as a child of screen.
- Sets the title of screen2 to "decryption".
- Sets the dimensions of screen2 to 400x200 pixels.
- Sets the background color of screen2 to #00bd56.

python
```
    message=text1.get(1.0,END)
    decode_message=message.encode("ascii")
    base64_bytes=base64.b64decode(decode_message)
```

```
    decrypt=base64_bytes.decode("ascii")
```

- Retrieves the text from the text1 widget.
- Encodes the retrieved text to ASCII format.
- Decodes the encoded message using Base64.
- Decodes the resulting Base64 bytes back to ASCII.

python
```
    Label(screen2,text="DECRYPT",font="arial",fg="white",bg="#00bd56").place(x=10,y=0)
    text2=Text(screen2,font="Rpbote 10",bg="white",relief=GROOVE,wrap=WORD,bd=0)
    text2.place(x=10,y=40,width=380,height=150)
    text2.insert(END,decrypt)
```

- Creates a label on screen2 with the text "DECRYPT".
- Creates a text widget text2 on screen2.
- Positions text2 on screen2 and inserts the decrypted message into it.

python
```
  elif password=="":
    messagebox.showerror("encryption","Input Password")
```

- Checks if no password was entered and shows an error message box.

python
```
  elif password !="1234":
    messagebox.showerror("encryption","Invalid Password")
```

- Checks if the entered password is incorrect and shows an error message box.

Encryption Function

python
```
def encrypt():
  password=code.get()
```

- def encrypt(): Defines the encrypt function, which will handle the encryption process.
- password=code.get(): Retrieves the entered password from the Entry widget associated with the code variable.

python
```
  if password=="1234":
```

- Checks if the entered password is "1234".

python
```
    screen1=Toplevel(screen)
    screen1.title("encryption")
```

```
    screen1.geometry("400x200")
    screen1.configure(bg="#ed3833")
```

- Creates a new window screen1 as a child of screen.
- Sets the title of screen1 to "encryption".
- Sets the dimensions of screen1 to 400x200 pixels.
- Sets the background color of screen1 to #ed3833.

python
```
    message=text1.get(1.0,END)
    encode_message=message.encode("ascii")
    base64_bytes=base64.b64encode(encode_message)
    encrypt=base64_bytes.decode("ascii")
```

- Retrieves the text from the text1 widget.
- Encodes the retrieved text to ASCII format.
- Encodes the ASCII text using Base64.
- Decodes the resulting Base64 bytes back to ASCII.

python
```
    Label(screen1,text="ENCRYPT",font="arial",fg="white",bg="#ed3833").place(x=10,y=0)
    text2=Text(screen1,font="Rpbote 10",bg="white",relief=GROOVE,wrap=WORD,bd=0)
    text2.place(x=10,y=40,width=380,height=150)
    text2.insert(END,encrypt)
```

- Creates a label on screen1 with the text "ENCRYPT".
- Creates a text widget text2 on screen1.
- Positions text2 on screen1 and inserts the encrypted message into it.

python
```
  elif password=="":
    messagebox.showerror("encryption","Input Password")
```

- Checks if no password was entered and shows an error message box.

python
```
  elif password !="1234":
    messagebox.showerror("encryption","Invalid Password")
```

- Checks if the entered password is incorrect and shows an error message box.

Main Screen Function

python
```
def main_screen():
  global screen
  global code
```

```
  global text1
```

- def main_screen(): Defines the main_screen function, which sets up the main user interface.
- global screen, code, text1: Declares screen, code, and text1 as global variables to be used in other functions.

python
```
  screen=Tk()
  screen.geometry("375x398")
  image_icon=PhotoImage(file="key.png")
  screen.iconphoto(False,image_icon)
  screen.title("pctApp")
```

- Creates the main application window screen.
- Sets the window size to 375x398 pixels.
- Loads an image from the file "key.png" and sets it as the window icon.
- Sets the title of the window to "pctApp".

python
```
  def reset():
    code.set("")
    text1.delete(1.0,END)
```

- Defines the reset function to clear the password entry and text area.

python
```
  Label(text="Enter text for encryption and
decryption",fg="black",font=("calbri",13)).place(x=10,y=10)
  text1=Text(font="Robote 20",bg="white",relief=GROOVE,wrap=WORD,bd=0)
  text1.place(x=10,y=50,width=355,height=100)
```

- Adds a label prompting the user to enter text for encryption/decryption.
- Creates a text widget text1 for user input and places it on the main window.

python
```
  Label(text="Enter secret key for encryption and
decryption",fg="black",font=("calbri",13)).place(x=10,y=170)
  code=StringVar()
  Entry(textvariable=code,width=19,bd=0,font=("arial",25),show="*").place(x=10,y=200)
```

- Adds a label prompting the user to enter a secret key.
- Creates a StringVar object code to store the password.
- Creates an entry widget for password input, associated with code, and places it on the main window.

python
```
Button(text="ENCRYPT",height="2",width=23,bg="#ed3833",fg="white",bd=0,command=enc
rypt).place(x=10,y=250)
```

Button(text="DECRYPT",height="2",width=23,bg="#00bd56",fg="white",bd=0,command=decrypt).place(x=200,y=250)

Button(text="RESET",height="2",width=50,bg="#1089ff",fg="white",bd=0,command=reset).place(x=10,y=300)

- Adds buttons for "ENCRYPT", "DECRYPT", and "RESET" with associated functions and places them on the main window.

python
```
screen.mainloop()
```

- Starts the Tkinter event loop, which waits for user interaction.

Running the Application

python
```
main_screen()
```

- Calls the main_screen function to run the application.

The code creates a basic GUI application using Tkinter that allows the user to enter text and a secret key. Depending on the key, the text can be encrypted or decrypted using Base64 encoding. The application displays the encrypted or decrypted message in a new window.