PGPTOOLS开发文档

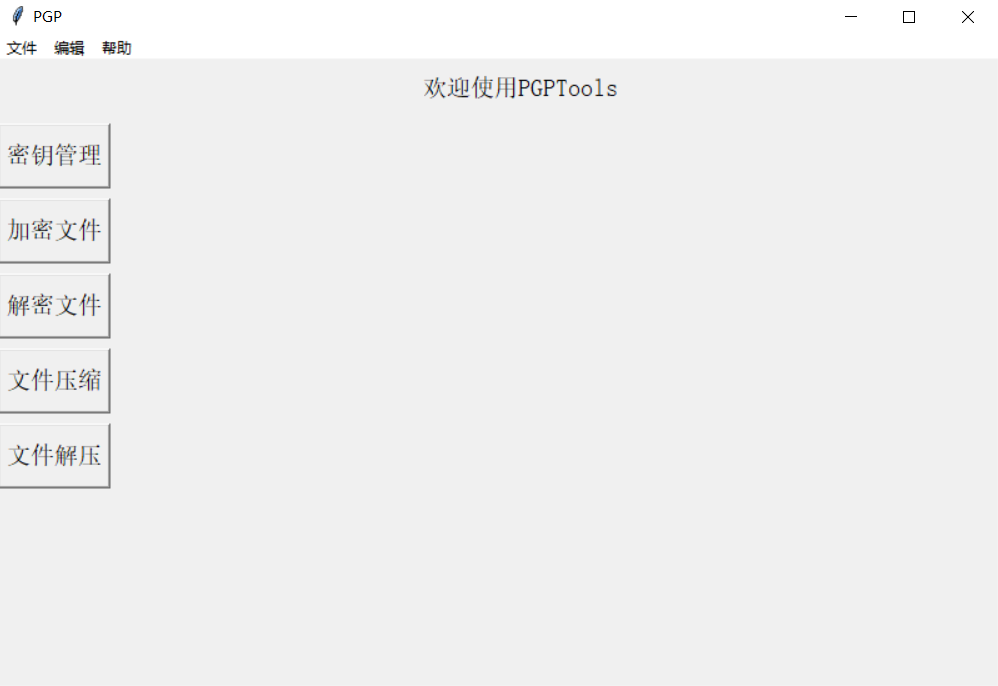
刘国翔

1. 前言

想哭了，之前写的文档在git重置后就没了。。。哭了。。。呜呜，我就简短地再写一遍吧。。

1. 功能构想
2. 密钥管理
3. 创建密钥（为私钥设置口令）
4. 删除密钥
5. 备份密钥
6. 加密文件
7. 解密文件
8. 压缩文件
9. 解压文件
10. 界面布局

初始界面



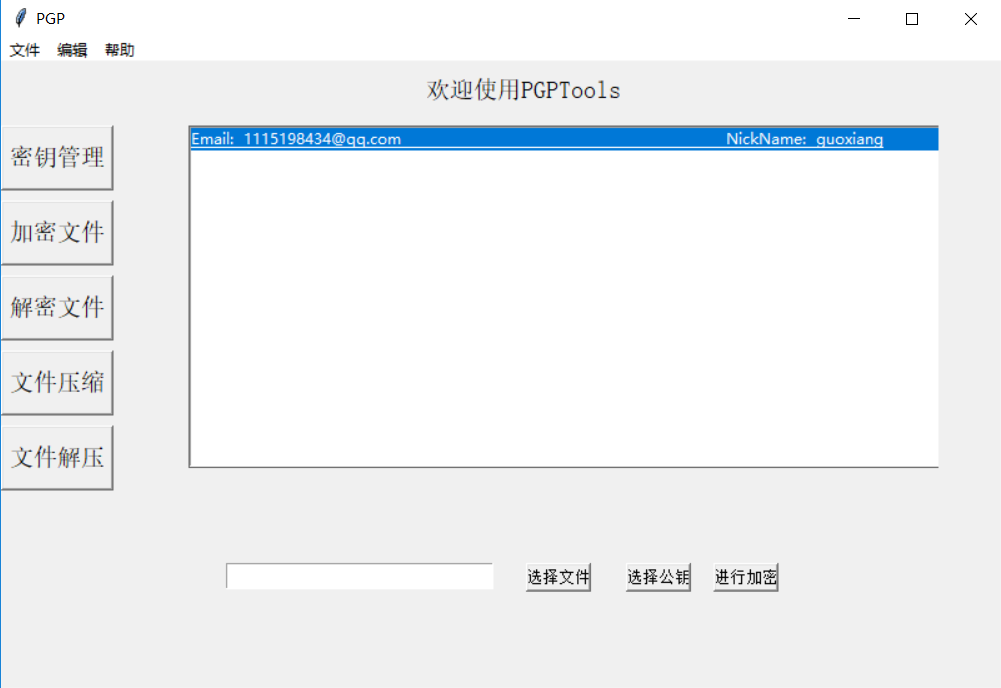
密钥管理界面



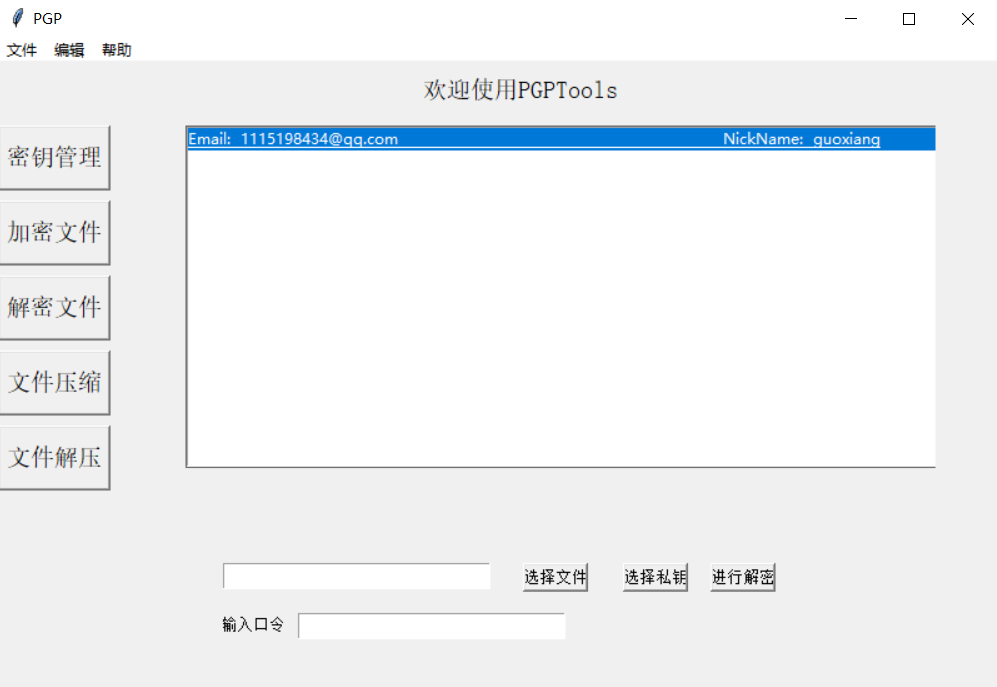
创建密钥界面



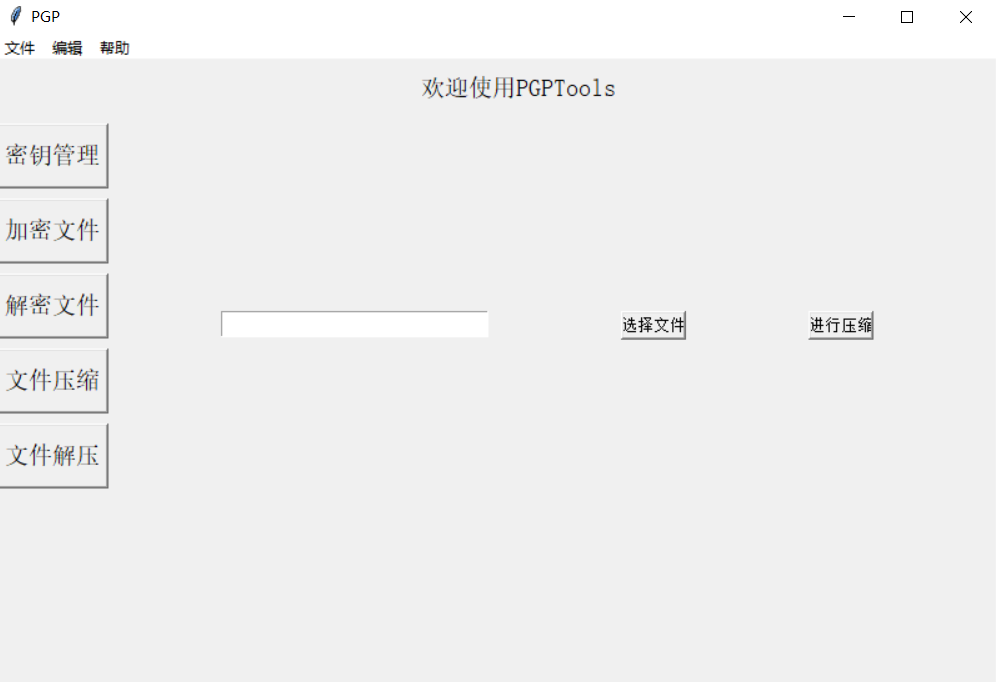
加密文件界面



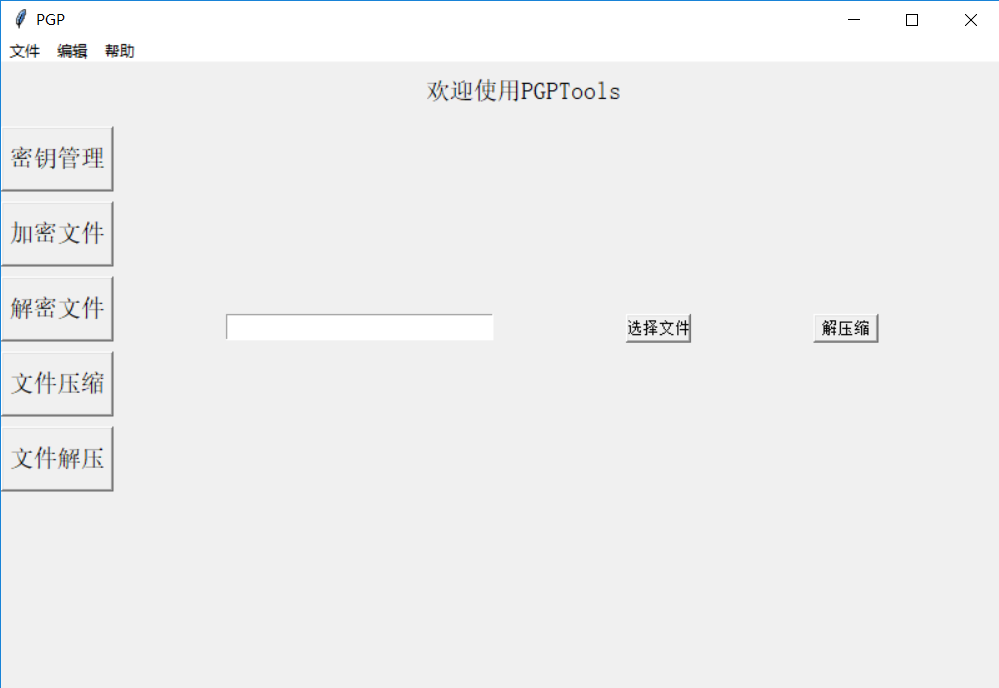
解密文件界面



文件压缩界面



文件解压界面



1. 代码实现

import tkinter as tk  
from tkinter import filedialog  
from pgpy.constants import PubKeyAlgorithm, KeyFlags, HashAlgorithm, SymmetricKeyAlgorithm, CompressionAlgorithm  
import pgpy  
import zlib  
  
window=tk.Tk()  
window.title('PGP')  
#window.iconbitmap('timg.jpg')  
window.geometry('800x500')  
#下面三个记录了被加密、解密和压缩文件的次序，暂且用来被当作加解密文件的文件名的区分，后续想想怎么存储一个被加密和被解密后的文件的名字  
secrectedFileNum=0  
decrptedFileNum=0  
compressedFileNum=0  
decompressedFileNum=0  
#存储密钥信息（如昵称、邮箱）的列表  
secrectNickNameContent=[]  
secrectEmailNmaeContent=[]  
secrectNum=0  
secrectListPublicFileName=''#被选中的公钥的文件名  
secrectListPrivateFileName=''#被选中的私钥的文件名  
file\_path=''#被选中的要进行加密的文件路径  
#菜单栏设置  
menubar=tk.Menu(window)  
#文件菜单栏设置  
filemenu=tk.Menu(menubar,tearoff=0)  
menubar.add\_cascade(label='文件',menu=filemenu)  
filemenu.add\_command(label='新建')  
filemenu.add\_command(label='打开')  
filemenu.add\_command(label='保存')  
filemenu.add\_command(label='退出',command=window.quit())  
#编辑菜单栏设置  
editMenu=tk.Menu(menubar,tearoff=0)  
menubar.add\_cascade(label='编辑',menu=editMenu)  
editMenu.add\_command(label='剪切')  
editMenu.add\_command(label='复制')  
editMenu.add\_command(label='粘贴')  
#帮助菜单栏设置  
helpMenu=tk.Menu(menubar,tearoff=0)  
menubar.add\_cascade(label='帮助',menu=helpMenu)  
helpMenu.add\_command(label='联系开发人员')  
#创建三级菜单  
submenu = tk.Menu(filemenu) # 和上面定义菜单一样，不过此处实在File上创建一个空的菜单  
filemenu.add\_cascade(label='导入', menu=submenu, underline=0) # 给放入的菜单submenu命名为Import  
# 第9步，创建第三级菜单命令，即菜单项里面的菜单项里面的菜单命令（有点拗口，笑~~~）  
submenu.add\_command(label='Submenu\_1') # 这里和上面创建原理也一样，在Import菜单项中加入一个小菜单命令Submenu\_1  
# 创建菜单栏完成后，配置让菜单栏menubar显示出来  
window.config(menu=menubar)  
#左边菜单栏的具体展开即其触发事件  
#界面切换管理的具体界面  
manageFrame=tk.Frame(window,height=500,width=700)  
#文件加密界面  
  
#h=tk.Label(manageFrame, text='on the frame\_l1', bg='green')  
#创建新密钥时所产生的新的对话框，要求用户输入昵称、邮件等  
def newDial():  
 def closeWin():  
 #实现弹出的创建框的关闭  
 secrectNickNameContent.append(nickName.get())  
 secrectEmailNmaeContent.append(email.get())  
 global secrectNum  
 secrectNum+=1  
 for i in range(secrectNum-1,secrectNum):  
 secretListContent.insert(tk.END, "Email:"+" "+secrectEmailNmaeContent[i]+" "+"NickName:"+" "+secrectNickNameContent[i])  
 #创建私钥、公钥  
 # we can start by generating a primary key. For this example, we'll use RSA, but it could be DSA or ECDSA as well  
 key = pgpy.PGPKey.new(PubKeyAlgorithm.RSAEncryptOrSign, 4096)  
  
 # we now have some key material, but our new key doesn't have a user ID yet, and therefore is not yet usable!  
 uid = pgpy.PGPUID.new(nickName.get(), comment='Honest Abe', email=email.get())  
 # now we must add the new user id to the key. We'll need to specify all of our preferences at this point  
 # because PGPy doesn't have any built-in key preference defaults at this time  
 # this example is similar to GnuPG 2.1.x defaults, with no expiration or preferred keyserver  
 key.add\_uid(uid, usage={KeyFlags.Sign, KeyFlags.EncryptCommunications, KeyFlags.EncryptStorage},  
 hashes=[HashAlgorithm.SHA256, HashAlgorithm.SHA384, HashAlgorithm.SHA512, HashAlgorithm.SHA224],  
 ciphers=[SymmetricKeyAlgorithm.AES256, SymmetricKeyAlgorithm.AES192, SymmetricKeyAlgorithm.AES128],  
 compression=[CompressionAlgorithm.ZLIB, CompressionAlgorithm.BZ2, CompressionAlgorithm.ZIP,  
 CompressionAlgorithm.Uncompressed])  
 #对私钥进行口令加密  
 key.protect(str(passWord.get()), SymmetricKeyAlgorithm.AES256, HashAlgorithm.SHA256)  
 #已创建私钥、公钥，现准备将其写入文件中  
 pubkey = key.pubkey  
 privateFileName=nickName.get()+" "+email.get()+" "+'privateKey'  
 publicFileName=nickName.get()+' '+email.get()+' '+'publicKey'  
 prikeyFile = open(privateFileName, 'w')  
 prikeyFile.write(str(key))  
 prikeyFile.close()  
 pubkeyFile = open(publicFileName, 'w')  
 pubkeyFile.write(str(pubkey))  
 pubkeyFile.close()  
  
  
 #print(nickName.get())#获取输入框里的内容  
 # print(email.get())#获取输入框里的内容  
 DialWin.destroy()  
 DialWin=tk.Tk()  
 DialWin.title('CreatSercet')  
 DialWin.geometry('400x300')  
 vNickName=tk.StringVar()  
 vEmail=tk.StringVar()  
 vPassWord=tk.StringVar()  
 tk.Label(DialWin,text='创建你自己的密钥',fg='black',font=('宋体',22),width=25,height=2).place(x=0,y=20)  
 tk.Label(DialWin, text='昵称', fg='black', font=('宋体', 12), width=10, height=2).place(x=50, y=82)  
 tk.Label(DialWin, text='邮箱', fg='black', font=('宋体', 12), width=10, height=2).place(x=50, y=132)  
 tk.Label(DialWin,text='口令',fg='black',font=('宋体',12),width=10,height=2).place(x=50,y=182)  
 nickName=tk.Entry(DialWin,textvariable=vNickName)  
 email=tk.Entry(DialWin,textvariable=vEmail)  
 passWord=tk.Entry(DialWin,textvariable=vPassWord)  
 nickName.place(x=150,y=90)  
 email.place(x=150,y=140)  
 passWord.place(x=150,y=190)  
 tk.Button(DialWin,text="创建",font=('宋体',14),width=8,height=2,command=closeWin).place(x=150,y=240)  
 DialWin.mainloop()  
  
  
#密钥管理界面控件的判断属性  
on\_hit=False  
#加密文件界面控件的判断属性  
on\_hit1=False  
#解密文件界面控件的判断属性  
on\_hit2=False  
#压缩文件界面控件的判断属性  
on\_hit3=False  
#解压文件界面控件的判断属性  
on\_hit4=False  
  
  
#创建密钥按钮  
miyaoCreatButton=tk.Button(manageFrame,text="创建密钥",font=('宋体',14),width=8,height=2,command=newDial)  
#密钥选择列表的相关选项  
#创建密钥选择列表  
def triggerOnLS(event):#鼠标键盘之类的触发事件；event要注意，否则报错  
 #对已经选择的密钥名进行处理，因为我在这里双击获取的密钥名的格式如下所示：Email: 1大赛爱的 NickName: 撒地方  
 secrectListSelected=secretListContent.get(secretListContent.curselection())  
 secrectListDealed=secrectListSelected.split(' ')  
 global secrectListPublicFileName  
 global secrectListPrivateFileName  
 secrectListPublicFileName=secrectListDealed[34]+' '+secrectListDealed[1]+" "+'publicKey'#获取了被选中的公钥的文件名  
 secrectListPrivateFileName=secrectListDealed[34]+' '+secrectListDealed[1]+' '+'privateKey'#获取了被选中的私钥的文件名  
 print(secrectListPublicFileName)  
 print(secrectListPrivateFileName)  
 #print(secrectListDealed[1])#获取了列表中被选中的密钥的Email名  
 #print(secrectListDealed[34])#获取了列表中被选中的密钥的NickName的名字  
 print(secretListContent.get(secretListContent.curselection()))  
secretListContent=tk.Listbox(manageFrame,width=100,height=15)  
secretListContent.bind('<Double-Button-1>',triggerOnLS)#获取键盘的命令后，执行相关的函数  
#删除密钥按钮  
miyaoDeleteButton=tk.Button(manageFrame,text="删除密钥",font=('宋体',14),width=8,height=2)  
#备份密钥按钮  
miyaoProtestButton=tk.Button(manageFrame,text="备份密钥",font=('宋体',14),width=8,height=2)  
  
  
  
  
  
  
#选择文件按钮的监听函数  
#选择的文件的绝对路径名的变量  
fileName=tk.StringVar()#相当于一个对象类型啦  
#在进行文件解密时获取输入的口令内容  
passWordContent=tk.StringVar()  
def selectFile():  
 global file\_path  
 file\_path=filedialog.askopenfilename()  
 fileName.set(file\_path)  
#选择文件的按钮  
fileChosenButton=tk.Button(manageFrame,text='选择文件',font=('宋体',10),width=6,height=1,command=selectFile)  
#选择文件后文件路径名的显示框  
fileNameEntry=tk.Entry(manageFrame,textvariable=fileName,width=30)  
#输入解密时的口令  
passWordEntry=tk.Entry(manageFrame,textvariable=passWordContent,width=30)  
#确定加密密钥的按钮  
fileSecrectQuerenButton=tk.Button(manageFrame,text='选择公钥',font=('宋体',10),width=6,height=1)  
#确定解密密钥的按钮  
fileDecrptQuerenButton=tk.Button(manageFrame,text='选择私钥',font=('宋体',10),width=6,height=1)  
#标题“输入口令”  
passWordLabel=tk.Label(manageFrame,text='输入口令',fg='black',font=('宋体',10),width=6,height=1)  
#加密文件函数,对‘进行加密’这个按钮的监听按钮  
def secrectFile():  
 global secrectedFileNum  
 secrectedFileNum+=1  
 secrectPubKey,\_=pgpy.PGPKey.from\_file(secrectListPublicFileName)  
 #要加密的文件的内容  
 file\_object = open(file\_path)  
 all\_the\_text = file\_object.read()  
 msg = pgpy.PGPMessage.new(all\_the\_text)  
 #利用已选中的公钥对文件进行加密  
 enc\_msg = secrectPubKey.encrypt(msg)  
 #将文件内容写到新文件中  
 dealedFile\_path=file\_path.split('.')  
 new\_file\_path='D:/PGPSecrectedFile/secrectedFile'+str(secrectedFileNum)+'.'+str(dealedFile\_path[1])  
 new\_file\_object=open(new\_file\_path,'w')  
 new\_file\_object.write(str(enc\_msg))  
 new\_file\_object.close()  
 print('文件加密成功')  
 #print(enc\_msg)#打印加密后的文件内容  
#对选中的文件进行解密的具体函数实现/对“进行解密”这个按钮的监听函数  
def decrptSelectedFile():  
 passWordFans=passWordEntry.get()  
 global decrptedFileNum  
 decrptedFileNum+=1  
 secrectPriKey,\_=pgpy.PGPKey.from\_file(secrectListPrivateFileName)  
 #对输入的口令进行判断，利用try...catch块进行  
 try:  
 with secrectPriKey.unlock(passWordFans):  
 # 要解密的文件的内容的读写  
 file\_object = open(file\_path)  
 all\_the\_txt = file\_object.read()  
 # 从文件中读取了str类型的内容后，要对该内容进行解密需要采用以下方法，及先将已有的内容转为PGPMessage类型，注意采用的是from\_blob而不是new  
 beingDecrpted = pgpy.PGPMessage.from\_blob(all\_the\_txt)  
 # 利用已选中的私钥对文件进行解密  
 dec\_msg = secrectPriKey.decrypt(beingDecrpted)  
 # 下面将解密后的文件内容生成一个新的文件，写在一个新的文件夹里  
 dealedFile\_path = file\_path.split('.')  
 new\_file\_path = 'D:/PGPDecrptedFile/decrptedFile' + str(decrptedFileNum) + '.' + str(dealedFile\_path[1])  
 new\_file\_object = open(new\_file\_path, 'w')  
 new\_file\_object.write(str(dec\_msg.message))  
 new\_file\_object.close()  
 print('文件解密成功')  
 except:  
 print("您输入的口令不对！")  
  
  
 #print(dec\_msg.message)  
#文件压缩的内部实现函数  
def compressFile():  
 global compressedFileNum  
 compressedFileNum+=1  
 file\_object=open(file\_path,'rb')#rb表示要读写二进制文件，否则会报错：提示不能是str类型，要为bytes类型  
 all\_the\_txt=file\_object.read()  
 compressedContent=zlib.compress(all\_the\_txt)  
 file\_object.close()  
 #下面将压缩后的文件内容写入一个新文件中  
 new\_file\_path='D:/PGPCompressFile/compressedFile'+str(compressedFileNum)+'.'+'txt'  
 new\_file\_object=open(new\_file\_path,'wb')  
 new\_file\_object.write(compressedContent)  
 new\_file\_object.close()  
 print('文件压缩成功')  
#文件解压缩的内部实现函数  
def decompressFile():  
 global decompressedFileNum  
 decompressedFileNum+=1  
 file\_object=open(file\_path,'rb')  
 all\_the\_txt=file\_object.read()  
 decompressedContent=zlib.decompress(all\_the\_txt)  
 file\_object.close()  
 # 下面将解压缩后的文件内容写入一个新文件中  
 new\_file\_path = 'D:/PGPDecompressedFile/decompressedFile' + str(decompressedFileNum) + '.' + 'txt'  
 new\_file\_object = open(new\_file\_path, 'wb')  
 new\_file\_object.write(decompressedContent)  
 new\_file\_object.close()  
 print('文件解压缩成功')  
  
  
#确定要进行加密的按钮  
sureToSecrectButton=tk.Button(manageFrame,text='进行加密',font=('宋体',10),width=6,height=1,command=secrectFile)  
#确定要进行解密的按钮  
sureToDecrptButton=tk.Button(manageFrame,text='进行解密',font=('宋体',10),width=6,height=1,command=decrptSelectedFile)  
#确定要进行文件压缩的按钮  
sureToCompressButton=tk.Button(manageFrame,text='进行压缩',font=('宋体',10),width=6,height=1,command=compressFile)  
#确定要进行文件解压缩的按钮  
sureToDecompressButton=tk.Button(manageFrame,text='解压缩',font=('宋体',10),width=6,height=1,command=decompressFile)  
  
  
#欢迎标语  
tk.Label(manageFrame,text='欢迎使用PGPTools',fg='black',font=('宋体',14),width=25,height=2).place(x=240,y=0)  
  
manageFrame.pack()  
#压缩文件的界面  
def compressionFileUI():  
 global on\_hit3  
 if on\_hit3==False:  
 on\_hit3=True  
 fileChosenButton.place(x=450,y=200,anchor='nw')  
 sureToCompressButton.place(x=600,y=200,anchor='nw')  
 fileNameEntry.place(x=130,y=200,anchor='nw')  
 else:  
 on\_hit3=False  
 fileChosenButton.place\_forget()  
 sureToCompressButton.place\_forget()  
 fileNameEntry.place\_forget()  
#解压缩文件的界面  
def decompressFileUI():  
 global on\_hit4  
 if on\_hit4==False:  
 on\_hit4=True  
 fileChosenButton.place(x=450, y=200, anchor='nw')  
 sureToDecompressButton.place(x=600, y=200, anchor='nw')  
 fileNameEntry.place(x=130, y=200, anchor='nw')  
 else:  
 on\_hit4=False  
 fileChosenButton.place\_forget()  
 sureToDecompressButton.place\_forget()  
 fileNameEntry.place\_forget()  
  
  
  
#解密文件的界面  
def decrptFile():  
 global on\_hit2  
 if on\_hit2==False:  
 on\_hit2=True  
 fileChosenButton.place(x=370, y=400, anchor='nw')  
 fileDecrptQuerenButton.place(x=450, y=400, anchor='nw')  
 passWordLabel.place(x=130,y=440,anchor='nw')  
 passWordEntry.place(x=190,y=440,anchor='nw')  
 fileNameEntry.place(x=130, y=400, anchor='nw')  
 sureToDecrptButton.place(x=520, y=400, anchor='nw')  
 secretListContent.place(x=100, y=50, anchor='nw')  
 else:  
 on\_hit2 = False  
 fileDecrptQuerenButton.place\_forget()  
 sureToDecrptButton.place\_forget()  
 passWordLabel.place\_forget()  
 passWordEntry.place\_forget()  
 fileChosenButton.place\_forget()  
 secretListContent.place\_forget()  
 fileNameEntry.place\_forget()  
  
#加密文件的界面  
def secrectFile():  
 global on\_hit1  
 if on\_hit1==False:  
 on\_hit1=True  
 fileChosenButton.place(x=370,y=400,anchor='nw')  
 fileSecrectQuerenButton.place(x=450,y=400,anchor='nw')  
 fileNameEntry.place(x=130,y=400,anchor='nw')  
 sureToSecrectButton.place(x=520,y=400,anchor='nw')  
 secretListContent.place(x=100, y=50, anchor='nw')  
 else:  
 on\_hit1=False  
 fileSecrectQuerenButton.place\_forget()  
 sureToSecrectButton.place\_forget()  
 fileChosenButton.place\_forget()  
 secretListContent.place\_forget()  
 fileNameEntry.place\_forget()  
  
  
#管理密钥的界面  
def manage():  
 global on\_hit  
 if on\_hit==False:  
 on\_hit=True  
 miyaoCreatButton.place(x=130,y=400,anchor='nw')  
 miyaoDeleteButton.place(x=280,y=400,anchor='nw')  
 miyaoProtestButton.place(x=430,y=400,anchor='nw')  
 secretListContent.place(x=100,y=50,anchor='nw')  
 else:  
 miyaoCreatButton.place\_forget()  
 miyaoDeleteButton.place\_forget()  
 miyaoProtestButton.place\_forget()  
 secretListContent.place\_forget()  
 on\_hit=False  
  
#加密文件  
#界面左端菜单栏  
manageButton=tk.Button(window,text='密钥管理',font=('宋体',14),width=8,height=2,command=manage).place(x=0,y=50,anchor='nw')  
secretButton=tk.Button(window,text='加密文件',font=('宋体',14),width=8,height=2,command=secrectFile).place(x=0,y=110,anchor='nw',)  
decretButton=tk.Button(window,text='解密文件',font=('宋体',14),width=8,height=2,command=decrptFile).place(x=0,y=170,anchor='nw')  
zipButton=tk.Button(window,text='文件压缩',font=('宋体',14),width=8,height=2,command=compressionFileUI).place(x=0,y=230,anchor='nw')  
unzipButton=tk.Button(window,text='文件解压',font=('宋体',14),width=8,height=2,command=decompressFileUI).place(x=0,y=290,anchor='nw')  
  
  
  
  
  
window.mainloop()