学号: xxxxxxxxxxx 姓名： xxxxxxx

## 7 文件的加密与解密

# 1程序描述

编写一个简易文件加密与解密程序，实现对选定文件的加密与解密功能，参考实现界面如下：



其中，界面可以采用tkinter实现，加密算法可以采用现有的加密和解密算法标准库实现，有的算法比如AES等，需要指定密钥长度，可以选择64位、128位或256位；如果加密或解密算法不需要这一项，可以去掉。

源文件是指需要进行加密或者解密的文件；目标文件是指加密或解密得到的文件。

# 2程序设计

## 2.1设计思路

加密在Web密码传输、文件传输以及其他保密工作中极其重要。加密算法分为对称加密和非对称加密两种，文件加密同城使用对称加密算法，如AES、DES等。

Python语言中用于加密的库有很多，其中最常用的是Cryptodome。这个库包含了很多常用的对称、非对称的加密算法，使用方法也很简单。

使用PyQt5设计软件界面，加密算法选择AES、DES两种，用户可以自行选择。AES和DES加密都有CBC和ECB两种模式，用户可以选择加密模式。

AES和DES加密都需要密钥且对密钥长度有限制。在Python的Cryptodome库中的AES加密所需要的密钥字符串长度可以是16/24/32位，其对应为AES-128/AES-196/AES-256。DES加密密钥字符串长度只能是8位。使用CBC模式时候，需要输入偏移量，AES的偏移量为16位，DES的偏移量位8位，软件应当对用户输入的密钥和偏移量进行校验并在不满足要求时给出提示。

为增强程序的鲁棒性，应该在进行加密前对输入文件路径、输出路径、密钥等进行校验，无法通过校验时给出提示，防止程序卡死退出。因为是对文件加密，文件类型多种多样，读取和写入时使用二进制文件操作(wb, rb)，防止文件损坏。

程序分层设计，加密层为核心，视图层仅为界面，逻辑层为视图层和核心层的连接桥率，所有校验操作和界面交互在此完成。

加密后的文件以原文件名+.data格式命名，防止输入和输出为同一个目录时覆盖掉原文件。解密时，根据加密的得到的文件名进行解析的到后缀名，如果更改过加密后的文件名后缀导致解析失败则直接以decrypt+当前时间戳命名，并提示手动更改后缀。

## 2.2程序代码

代码见src文件夹

已经打包为可执行文件，见dist文件夹

代码运行

因使用了第三方库，运行前需要先安装依赖，依赖已经导出为requirements.txt，命令行窗口进入src目录后使用以下命令安装（使用清华大学镜像源，国内网络环境可直接在线安装）：

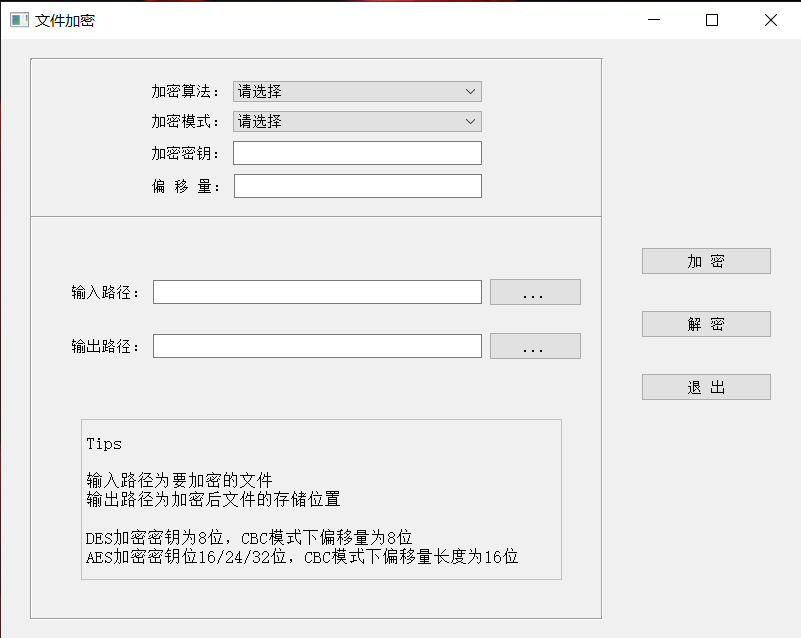
pip install -i https://mirrors.aliyun.com/pypi/simple/ -r requirements.txt

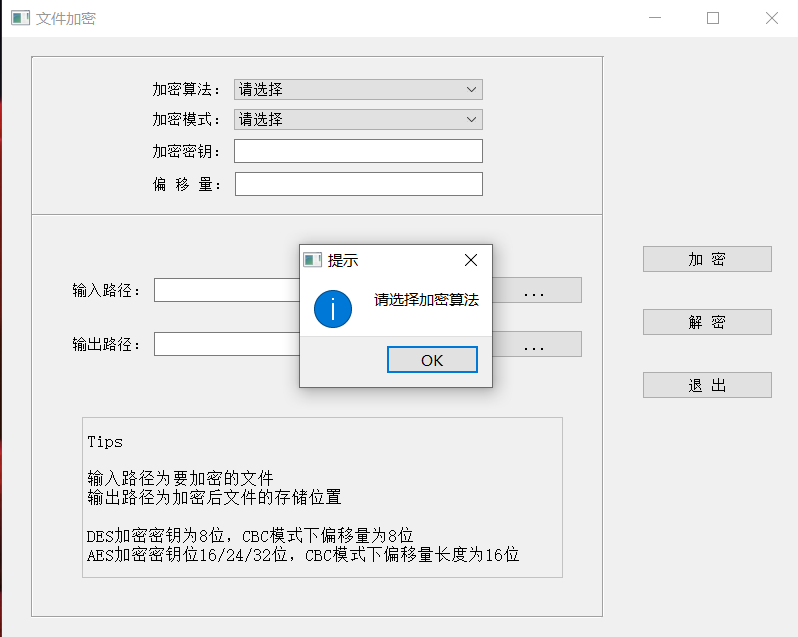
安装依赖后运行mian.py即可。

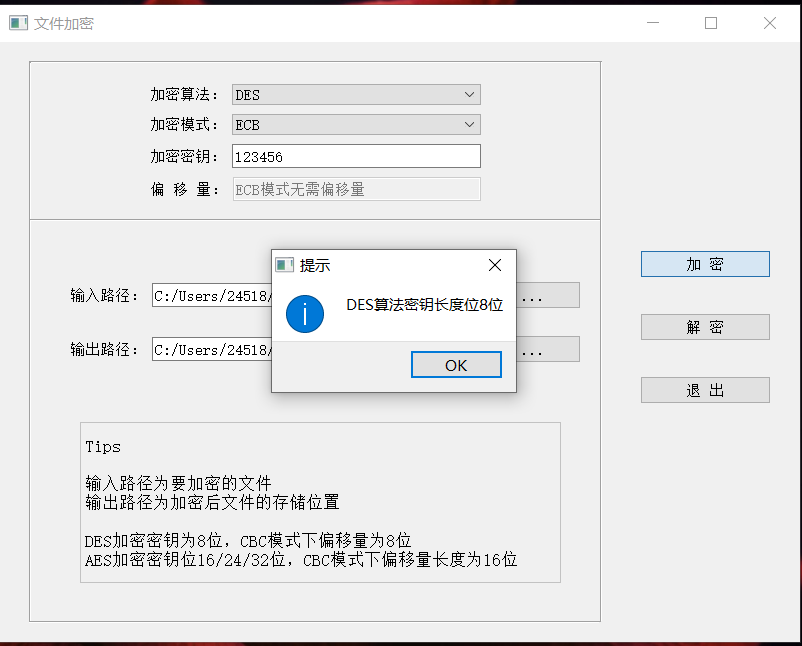
（程序代码与实验报告一起打包提交）

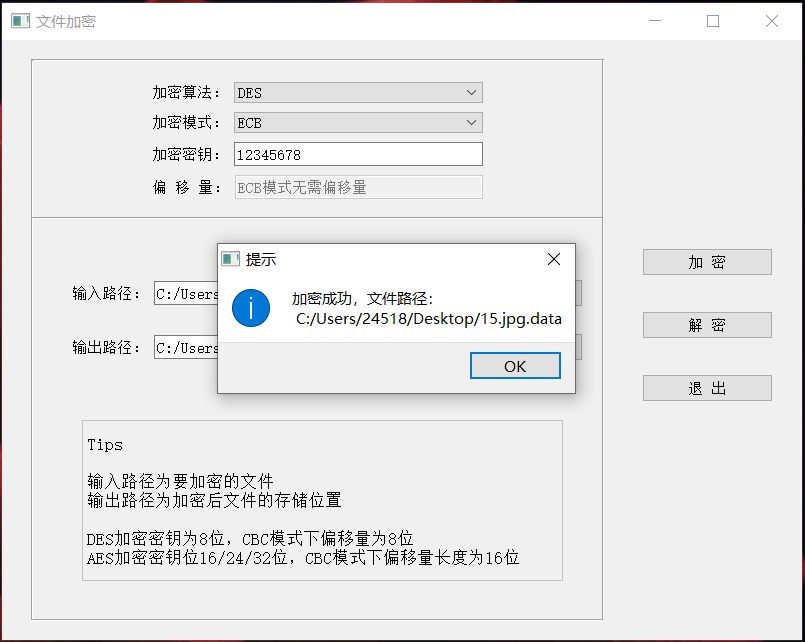
## 2.3运行结果

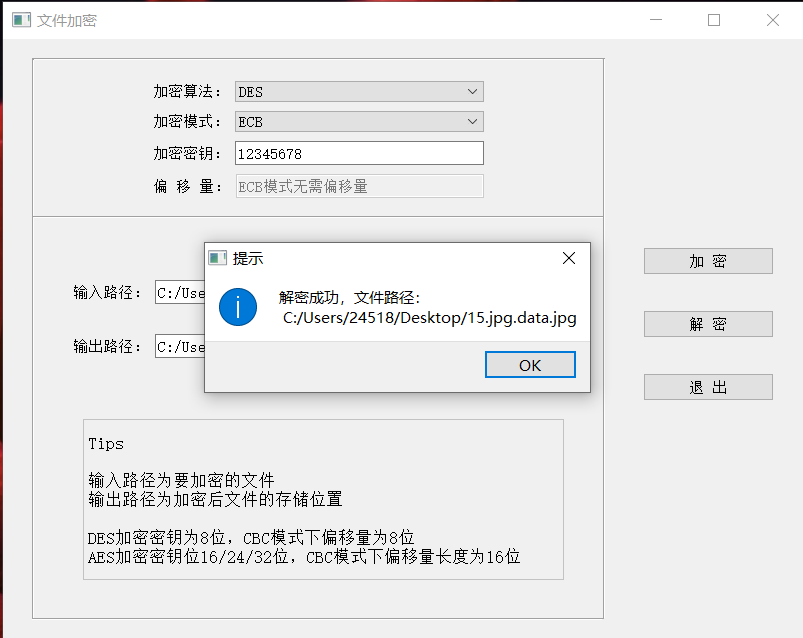
输入路径和输出路径旁边的按钮可以浏览选择文件/目录

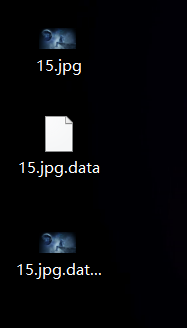












# 3总结

相比Tkinter，PyQt的槽函数机制更容易实现程序分层设计，视图和逻辑分开，此外，PyQt的控件比Tkinter的要美观一些。

Python的加密解密库很容易使用，而且功能强大，一个库几乎包含所有常用的加密算法。