第一次作业

1. 模拟PBE对文件进行加解密

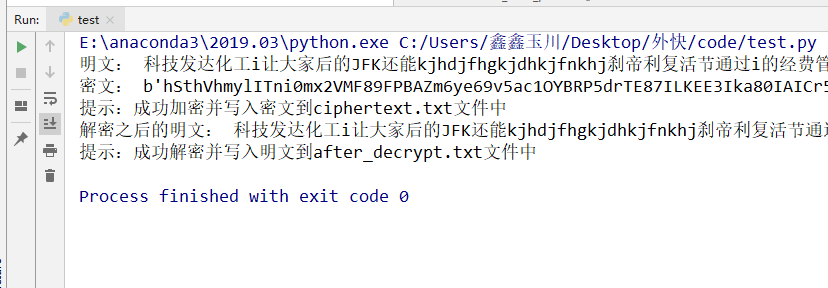
前言：

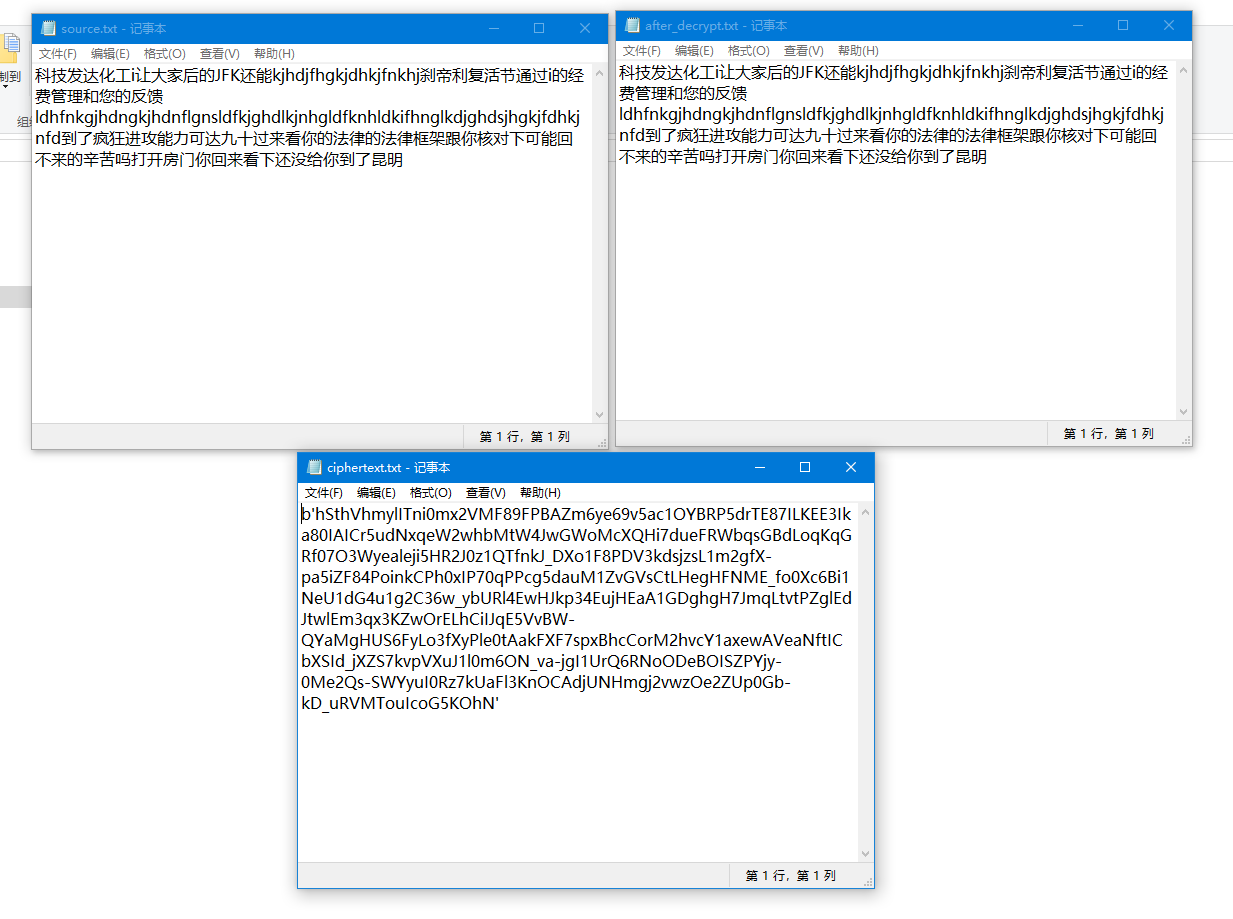
平台：pycharm，语言：python，加密的文件：source.txt，加密后的文件：ciphertext.txt，解密后的文件：after\_decrypt.txt

核心代码：详细代码在打包的文件中

|  |
| --- |
| # 功能：利用PBE创建私钥  # 参数：password是用户口令  # 返回值：返回创建的私钥  def create\_key(password):  # 功能：加密消息  # 参数：key为密钥 plain\_text是需要被加密的文本  # 返回值：加密后的base64码，AesGcm加密使用的once值  def encryption\_mesg(key,plain\_text):  # 功能: 解密消息  # 参数: cipher\_text是加密后的64base码  # 返回值：返回解密后的文本  def decryption\_mesg(cipher\_text,nonce,key):  # 主函数模拟加解密文件过程  def PBE\_total\_process():  password = "081849xx@zzq" #定义口令  key = create\_key(password) #获得密钥  plain\_text = openFile('source.txt') # 获得文本  cipher\_text,nonce = encryption\_mesg(key,plain\_text) # 进行加密  if (writeFile('ciphertext.txt',cipher\_text)):  print("成功加密并写入密文到ciphertext.txt文件中")  else:  print("加密过程出现错误")  decryption\_key = key # 简单模拟密钥传递  decrypted\_cipher\_text = decryption\_mesg(cipher\_text,nonce,decryption\_key)  if (writeFile('after\_decrypt.txt',decrypted\_cipher\_text)):  print("成功解密并写入明文到after\_decrypt.txt文件中")  else:  print("解密过程出现错误")  if \_\_name\_\_ == '\_\_main\_\_':  # demonstrate method  PBE\_total\_process() |

最终结果：

 程序输出为：

加密文件，密文文件，解密文件内容：