Project4：基于open PGP实现本地加密文件夹

2151140 王谦 信息安全

一、实验要求

> 一、设计一个本地文件处理协议，基于open PGP实现本地加密文件夹：
>
> 1.对目标文件实现对存储者和调阅者的基于pgp的真实性认证和文件加密；
>
> 2.上述文件安全性不依赖于本地系统，即
>
> a）本地其他非授权用户（即便是系统管理员）无法以可理解的方式读出该文件夹中文件内容；
>
> b）对处理过程中可能涉及的临时存储至少实现可靠的敏感信息残留覆盖；
>
> 二、在选择linux或MS windows，实现该协议的一个C++实现实例。包括软件设计文档、源代码及注释、可执行安装包、自测用例和测试分析报告、第三方资源及其说明。

二、open pgp 基础介绍

OpenPGP 加密过程包括以下步骤：

1. **随机生成一个对称密钥（Symmetric Key）**：使用对称加密算法生成一个随机密钥，这个密钥用于加密要传输的数据。
2. **使用对称加密算法加密数据**：使用上一步生成的对称密钥对要传输的数据进行加密。
3. **使用接收者的公钥加密对称密钥**：使用非对称加密算法（如 RSA）中接收者的公钥来加密刚刚生成的对称密钥。
4. **传输加密的数据和加密的对称密钥**：将加密后的数据和使用接收者的公钥加密过的对称密钥一起传输给接收者。

接收者收到加密的数据后，进行解密：

1. **使用自己的私钥解密对称密钥**：使用接收者的私钥来解密接收到的对称密钥，从而获得对称密钥。
2. **使用对称密钥解密数据**：使用上一步解密得到的对称密钥来解密接收到的加密数据，从而获取原始数据。

数字签名的原理包括以下步骤：

1. **生成数据的哈希值**：发送者对要传输的数据使用加密散列函数（如 MD5、SHA-256）生成数据的哈希值。
2. **使用发送者的私钥加密哈希值**：发送者使用自己的私钥对数据的哈希值进行加密，生成数字签名。
3. **传输数据和数字签名**：将原始数据和数字签名一起传输给接收者。

接收者收到数据后，进行验证：

1. **使用发送者的公钥解密数字签名**：接收者使用发送者的公钥来解密收到的数字签名，得到数据的哈希值。
2. **生成数据的哈希值**：接收者对收到的数据使用相同的加密散列函数生成数据的哈希值。
3. **比对哈希值**：接收者将解密得到的哈希值与自己重新计算的数据的哈希值进行比对。如果两者相同，则数字签名有效且数据完整。

三、功能分析

单用户授权流程：

1. 用户 A 想要存储文件，并保证只有自己能够访问。
2. 用户 A 对文件进行签名：使用 MD5 加密方式获得文件的哈希值，然后用用户 A 的私钥加密该哈希值，并将其附在文件末尾。
3. 用户 A 对文件进行加密：将带有签名的文件通过用户 A 的公钥加密，保存为.gpg 文件。

多用户授权流程：

1. 用户 A 想要存储文件，并保证自己和用户 B 都能够访问。
2. 用户 A 对文件进行签名：使用 MD5 加密方式获得文件的哈希值，然后用用户 A 的私钥加密该哈希值，并将其附在文件末尾。
3. 用户 A 对文件进行加密：将带有签名的文件通过用户 A 和用户 B 的公钥加密，保存为.gpg 文件。

本实验的功能说明：

1. 进入程序首先选择目录地址，随后会在目录地址创建一个OpenPGP_File_Manage_show文件夹。
2. 同时根据用户账户记录id，为用户创建密钥，并建立用户名子文件夹，密钥存在其中的Key文件夹。
3. 随后选择加密选项并选择单用户，输入加密文件位置，加密后会将加密的.pgp文件存到用户名子文件的File文件夹。
4. 随后可以按任意键返回菜单，然后选择解密选项，把待解密的.pgp文件位置输入，身份校验后成功在同目录下解密出原文件。

四、实验过程

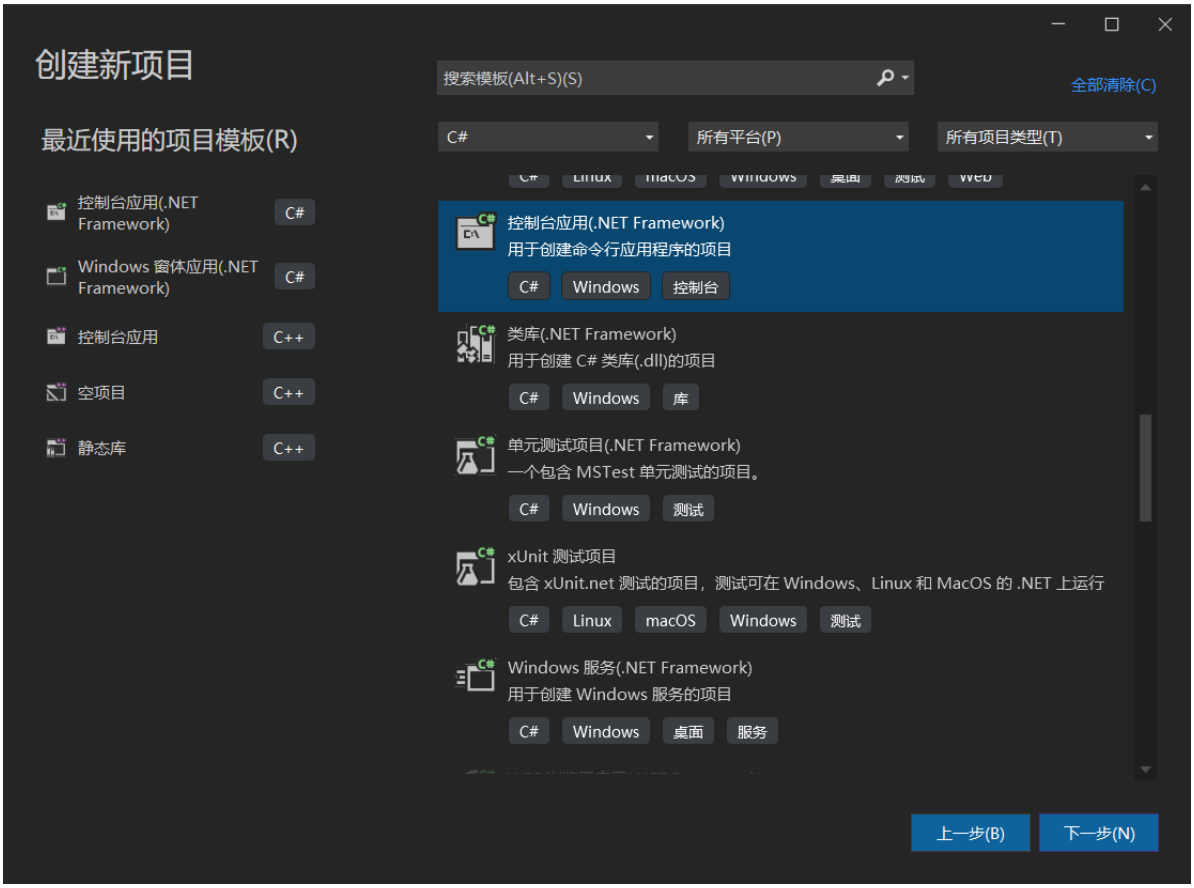
在网络上找到了学长学姐以前的成功案例，因此仿照着成功的先例用C#来进行尝试学习，而没有使用C++。

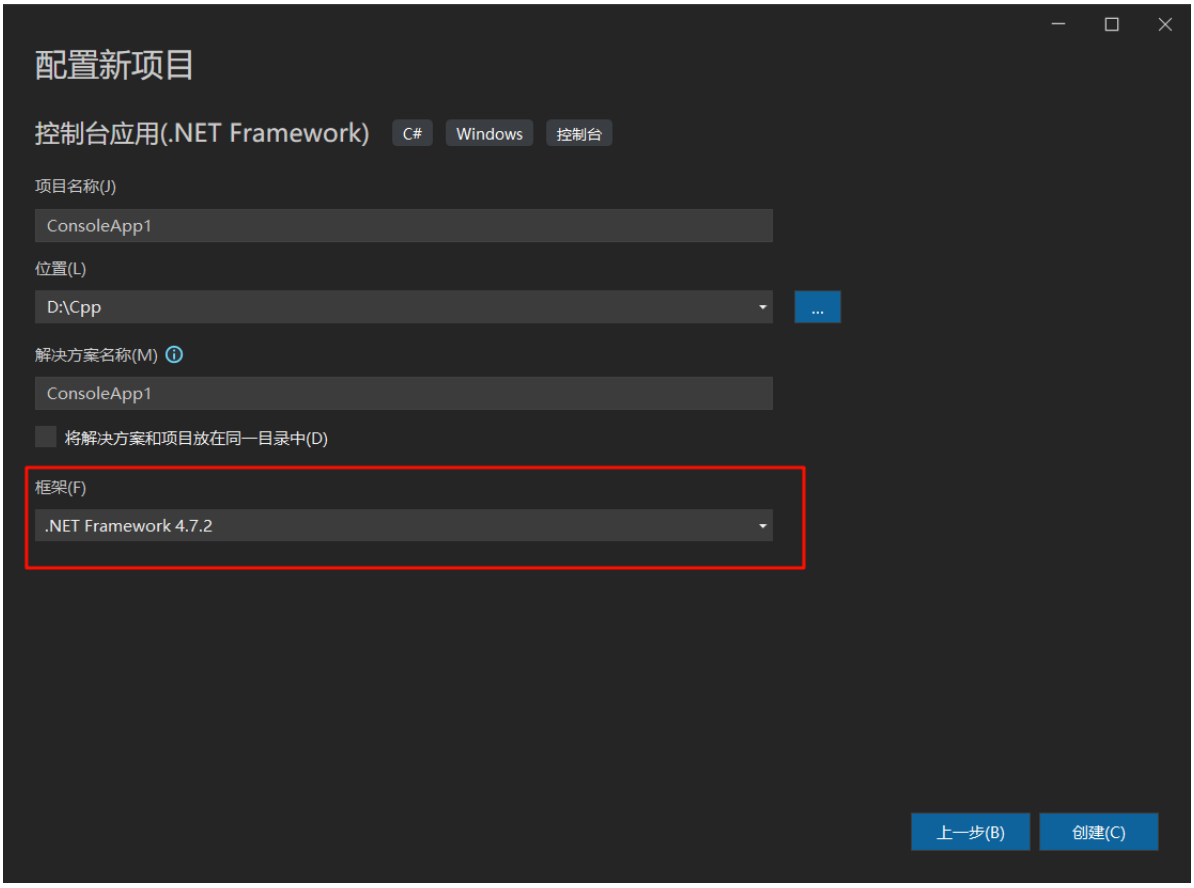
1.环境配置和代码

Windows10；Visual Studio 2022；C#控制台应用；.Net Framework 4.7.2；DidiSoft.Pgp.Trial（NuGet）

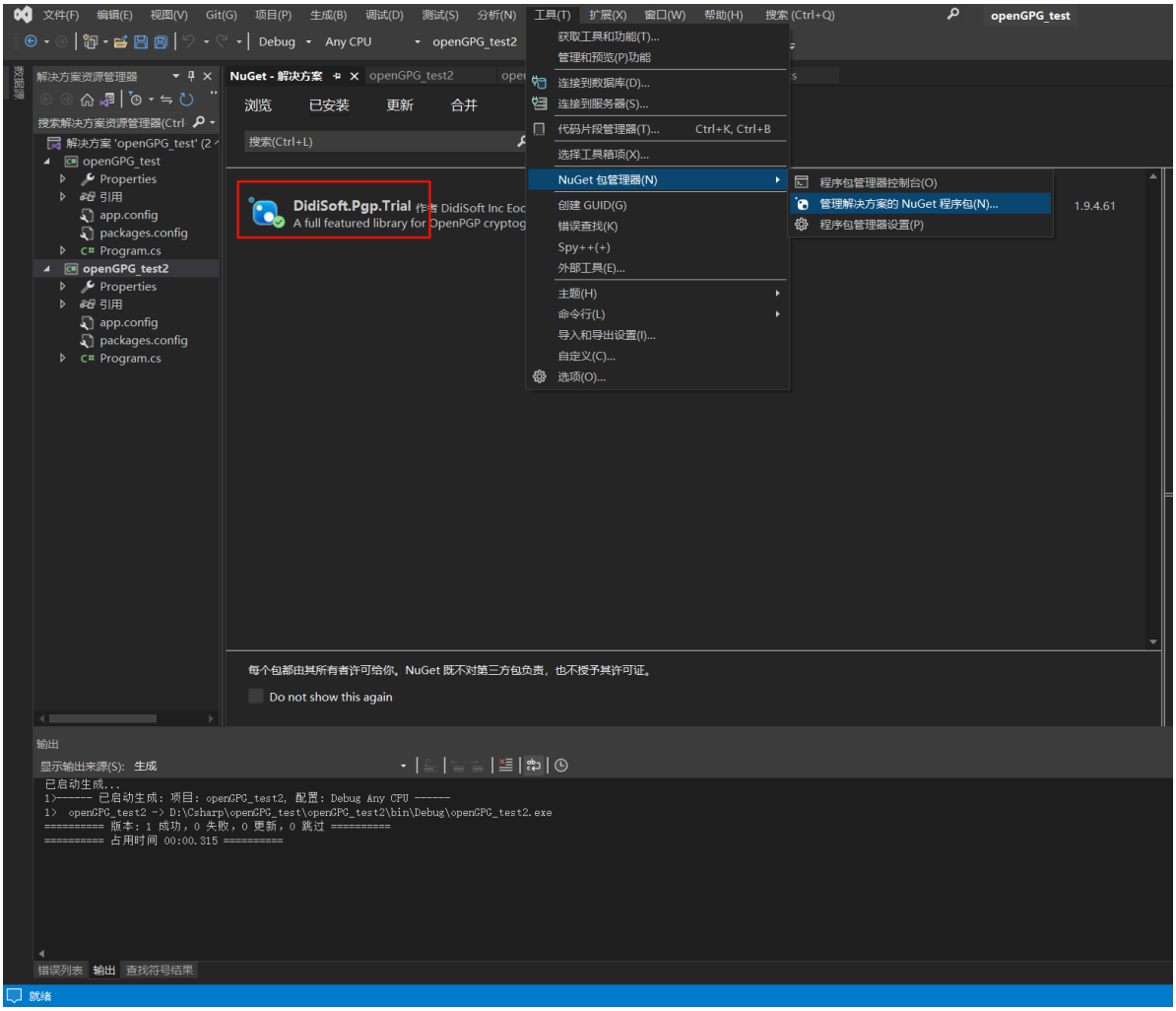
VS安装C#.NET相关组件

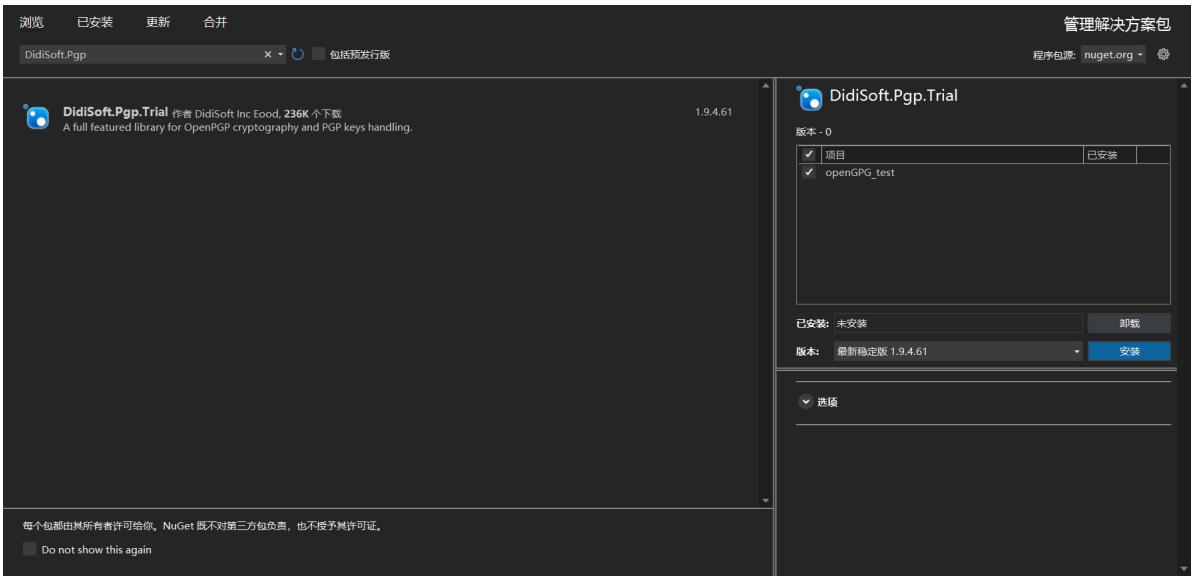
创建最简单的C#控制台项目，选择了.Net Framework 4.7.2





使用NuGet对项目安装DidiSoft.Pgp.Trial





通过DidiSoft.Pgp.Trial可以在C#中方便的实现OpenPGP的功能

OpenPGP Library for .NET

DidiSoft Inc | 2,812 clicks | (0) | Free Trial

DidiSoft OpenPGP Library for .NET offers simple but powerful API for Open PGP cryptography operations. You will find all the encryption and signing OpenPGP functions against Files, Strings, Streams, Key management and key servers communication.

Get Started

Overview Q & A Rating & Review

DidiSoft OpenPGP Library for .NET is a simple, yet powerful class library that offers Open PGP encryption functionality. The list of available features starts with the OpenPGP standard (RFC 4880 and RFC 6637) and a wide set of extra functionality like support for multiple files inside a .PGP archive and connectivity to LDAP and HKP PGP key servers.

The Example below illustrates OpenPGP encryption of a file:

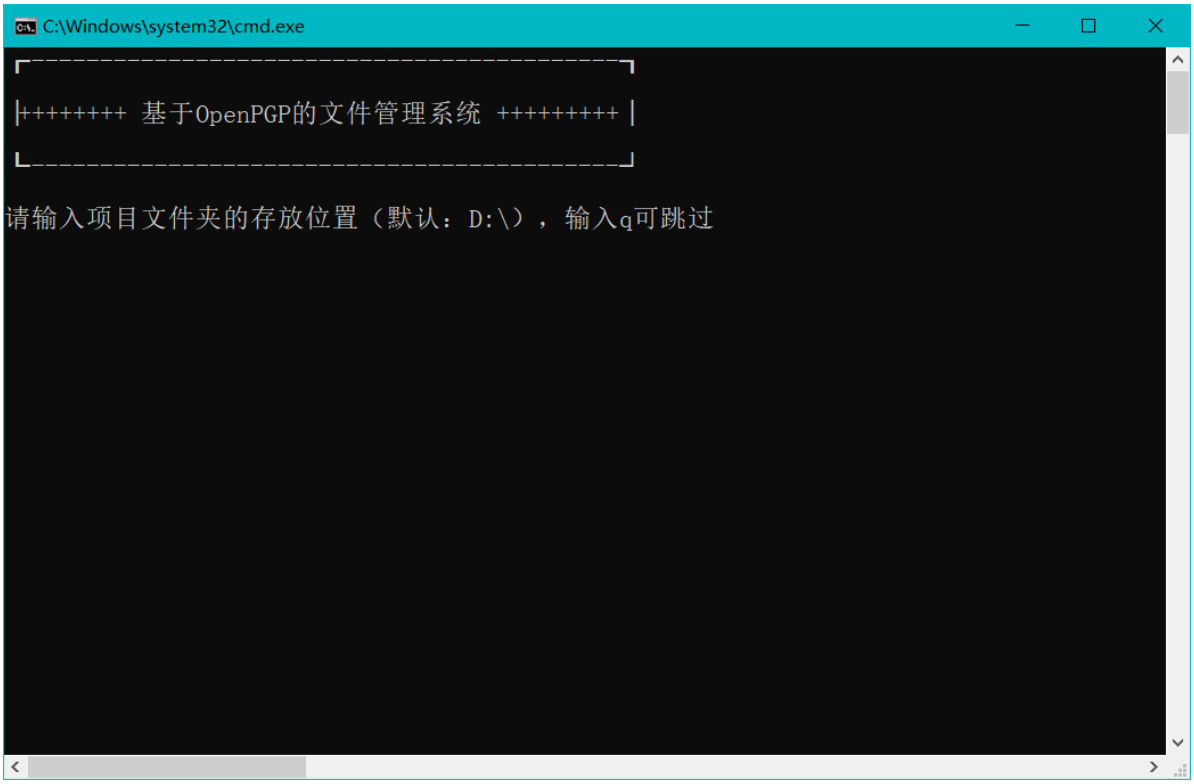
```
// C# sample
PGPLib pgp = new PGPLib();
// should output be ASCII or binary
bool asciiArmor = false;
// PGP encrypt
pgp.EncryptFile(@"C:\INPUT.txt", @"C:\recipient_public_key.asc", @"C:\OUTPUT.pgp", asciiArmor);


'-- VB.NET example
Dim pgp As New PGPLib()
' should output be ASCII or binary

Dim asciiArmor As Boolean = False
' encrypt
pgp.EncryptFile("C:\INPUT.txt", "C:\recipient_public_key.asc", "C:\OUTPUT.pgp", asciiArmor)
```

DidiSoft OpenPGP Library for .NET edition supports .NET Framework (4.6, 4.5.1, 4.5, 4.0, 3.5, 2.0), UWP, .NET Core 1.1, WinRT, Silverlight 5, .NET CF (3.5, 2.0) on Windows CE and Windows Mobile devices, Windows Phone 8.1 and Windows Phone 8, Xamarin.iOS or Xamarin.Android.

随后可以开始编写C#代码，代码大量参考了学长学姐的内容，将在后面给出代码的分析。

2.程序使用展示

打开程序首先输入创建文件夹的位置，默认为D:\



随后创建总文件夹和其下的密钥文件夹和用户文件夹

根据用户账户信息给出uid等标识，并计算密钥存储到Key文件夹

```
┌──────────────────────────────────────────┐
|++++++++ 基于OpenPGP的文件管理系统 +++++++++|
└──────────────────────────────────────────┘

请输入项目文件夹的存放位置（默认：D:\），输入q可跳过
q
应用所创建的文件夹信息
--------------------------------------------------------------------------------------------------------------
在"D:\OpenPGP_File_Manage_show" 创建了文件夹："基于OpenGPG的文件系统"的总文件夹
在"D:\OpenPGP_File_Manage_show\连阡" 创建了文件夹：用户连阡的用户文件夹
在"D:\OpenPGP_File_Manage_show\连阡\Key" 创建了文件夹：用户连阡的密钥（可以导出公钥）文件夹
在"D:\OpenPGP_File_Manage_show\连阡\File" 创建了文件夹：用户连阡的文件（加密、解密后的文件）的文件夹
--------------------------------------------------------------------------------------------------------------
当前用户信息
--------------------------------------------------------------------------------------------------------------
当前活动用户名为：连阡
安全标识符为：S-1-5-21-1315629951-545323187-4047727180-1004
当前userID为：-1616283012
--------------------------------------------------------------------------------------------------------------

生成用户密钥
--------------------------------------------------------------------------------------------------------------
用户连阡的密钥已存在，在"D:\OpenPGP_File_Manage_show\连阡\Key\key.store"中
用户连阡的公钥已导出，在"D:\OpenPGP_File_Manage_show\连阡\Key\public_key_exported.asc"中
当前密钥信息为：
Username        Type      Key Id                    Created           User Id
连阡              pub/sec   3050940519350670649         2024/4/24          -1616283012
--------------------------------------------------------------------------------------------------------------
原理展示请按1，存储文件请按2，调阅文件请按3
```

选择存储文件，如果选择多用户，可以检测到其他用户


```
当前密钥信息为：
Username        Type      Key Id                    Created           User Id
连阡              pub/sec   3050940519350670649         2024/4/24          -1616283012
--------------------------------------------------------------------------------------------------------------

原理展示请按1，存储文件请按2，调阅文件请按3
2

文件存储
--------------------------------------------------------------------------------------------------------------
存储文件的安全模式：仅自己请按1，多用户请按2
2
检测到用户liech
您是否要为用户liech开放该文件的调阅权限？（Y/N）
```

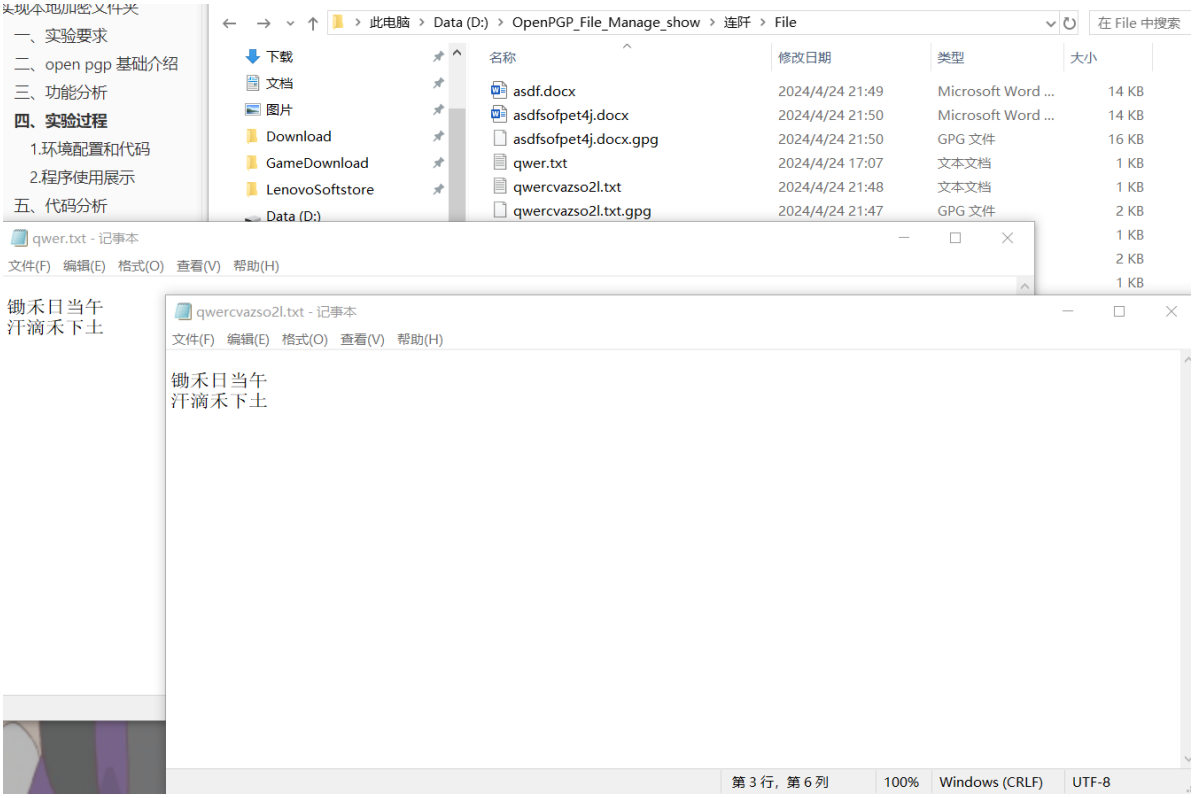
qwer.txt加密成功.pgp文件存到用户文件夹中

```
原理展示请按1，存储文件请按2，调阅文件请按3
2

文件存储
--------------------------------------------------------------------------------------------------------------
存储文件的安全模式：仅自己请按1，多用户请按2
2
检测到用户liech
您是否要为用户liech开放该文件的调阅权限？（Y/N）
y
为用户liech开放该文件的调阅权限成功
请输入文件路径
D:\OpenPGP_File_Manage_show\连阡\File\qwer.txt
身份认证成功，您的身份为连阡,创建文件成功
文件存储成功，并由用户"连阡"签名，在"D:\OpenPGP_File_Manage_show\连阡\File\qwercvazso2l.txt.gpg"中
--------------------------------------------------------------------------------------------------------------

程序已结束，按q退出，按其他任意键返回用户界面...

原理展示请按1，存储文件请按2，调阅文件请按3
3

文件调阅
--------------------------------------------------------------------------------------------------------------
请输入文件路径(后缀为.gpg)
D:\OpenPGP_File_Manage_show\连阡\File\qwercvazso2l.txt.gpg
2
您的身份为用户"连阡"
文件调阅成功，原文件名为qwer.txt,解密后的文件在D:\OpenPGP_File_Manage_show\连阡\File\qwercvazso2l.txt中
签名验证成功，该文件是由用户连阡创建的
--------------------------------------------------------------------------------------------------------------

程序已结束，按q退出，按其他任意键返回用户界面...
```

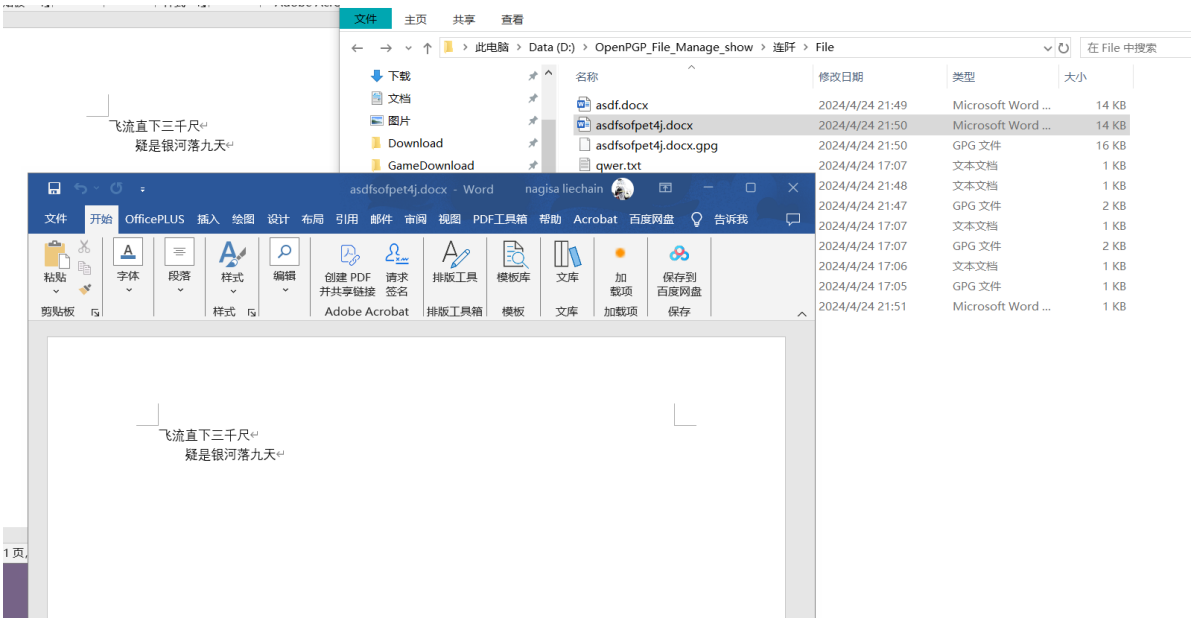
选择单用户存储asdf.docx文件，然后尝试解密

```
原理展示请按1，存储文件请按2，调阅文件请按3
2

文件存储
--------------------------------------------------------------------------------------------------------------
存储文件的安全模式：仅自己请按1，多用户请按2
1
该文件由用户连阡创建，并且只能由用户连阡查看
请输入文件路径
D:\OpenPGP_File_Manage_show\连阡\File\asdf.docx
身份认证成功，您的身份为连阡,创建文件成功
文件存储成功，并由用户"连阡"签名，在"D:\OpenPGP_File_Manage_show\连阡\File\asdfsofpet4j.docx.gpg"中
--------------------------------------------------------------------------------------------------------------

程序已结束，按q退出，按其他任意键返回用户界面...

原理展示请按1，存储文件请按2，调阅文件请按3
3

文件调阅
--------------------------------------------------------------------------------------------------------------
请输入文件路径(后缀为.gpg)
D:\OpenPGP_File_Manage_show\连阡\File\asdfsofpet4j.docx.gpg
2
您的身份为用户"连阡"
文件调阅成功，原文件名为asdf.docx,解密后的文件在D:\OpenPGP_File_Manage_show\连阡\File\asdfsofpet4j.docx中
签名验证成功，该文件是由用户连阡创建的
--------------------------------------------------------------------------------------------------------------

程序已结束，按q退出，按其他任意键返回用户界面...
```

qwer.txt成功解密



asdf.docx文件成功解密

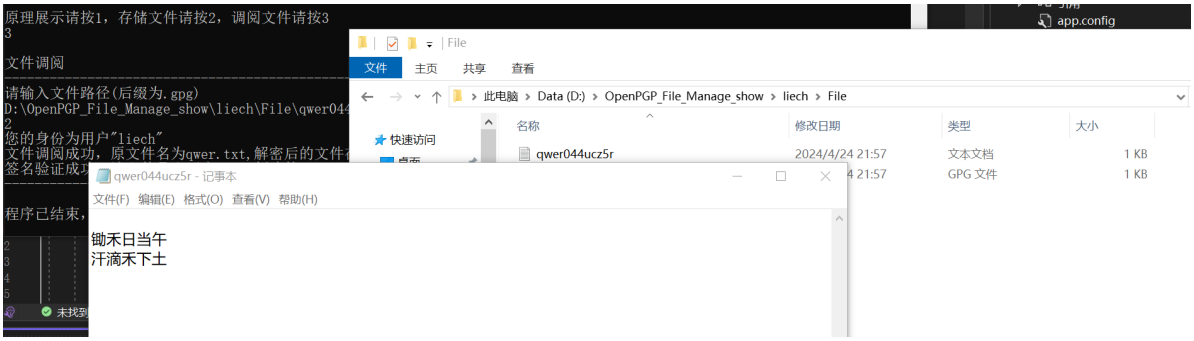


切换成另一个用户

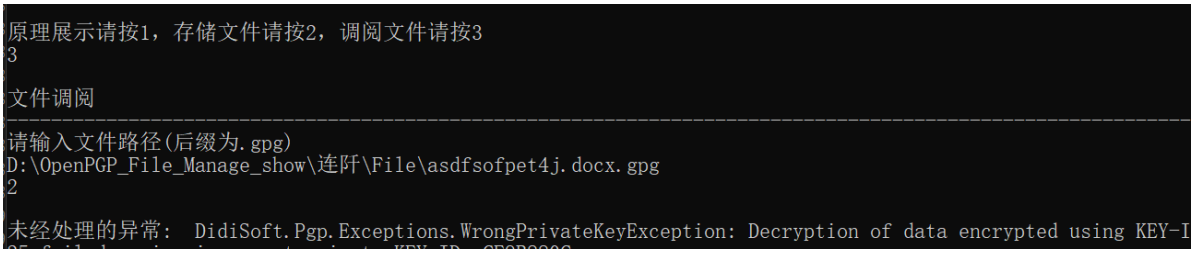
```
当前用户信息
-------------------------------------------------------------------------------
当前活动用户名为：liech
安全标识符为：S-1-5-21-1315629951-545323187-4047727180-1012
当前userID为：-1616283036
-------------------------------------------------------------------------------

生成用户密钥
-------------------------------------------------------------------------------
用户liech的密钥已存在，在"D:\OpenPGP_File_Manage_show\liech\Key\key.store"中

用户liech的公钥已导出，在"D:\OpenPGP_File_Manage_show\liech\Key\public_key_exported.asc"中

当前密钥信息为：
Username        Type      Key Id                  Created         User Id
liech           pub/sec   1771554582910699532     2024/4/24       -1616283036
-------------------------------------------------------------------------------
```

用户具备管理员权限

qwer.txt可以解密



asdf.docx无法解密



五、代码分析

User 类功能流程

1. **构造函数**：
    - 获取当前活动用户名。
    - 获取当前用户的 Windows 安全标识符。
    - 调用 `GetUserID()` 方法生成用户ID。

2. **GetUserID()**：
    - 使用 `PGPLib` 设置哈希算法为 MD5。
    - 将安全标识符的哈希值转换为字符串作为用户ID。
3. **User_Authorization()**：
    - 获取用户文件夹路径。
    - 检测是否存在其他用户。
    - 遍历所有用户，询问是否授权访问文件。
4. **detect_All_User_Public_key()**：
    - 获取用户文件夹路径。
    - 检测每个用户的公钥文件是否存在。
    - 构建已授权用户和公钥路径的列表。

Global 类功能流程

1. **构造函数**：
    - 初始化文件夹信息。
    - 根据用户输入设置文件基础路径。
    - 创建总文件夹、用户文件夹、密钥文件夹和文件文件夹。
2. **SetBaseName()**：
    - 询问用户输入项目文件夹存放位置。
    - 根据用户输入更新文件夹路径。

ModeManage 类功能流程

1. **mode_input(User user)**：
    - 显示操作模式选择菜单。
    - 根据用户输入选择模式并返回模式编号。
2. **mode_control(int mode, User user, string passwd)**：
    - 根据模式编号执行相应的文件操作：
        - 模式1：展示原理，加密字符串。
        - 模式21：存储文件，仅限自己访问。
        - 模式22：存储文件，多用户访问。
        - 模式3：调阅文件。

KeyManage 类功能流程

1. **GenerateKeyPairRSA(string userID, string passwd)**：
    - 检查密钥库文件是否存在，不存在则创建密钥库。
    - 生成RSA密钥对并存储在密钥库中。
2. **ExportPublicKey(string userID, string passwd)**：
    - 打开密钥库。
    - 导出公钥到指定文件。
3. **ExportPrivateKey(string userID, string passwd)**：
    - 打开密钥库。
    - 导出私钥到指定文件。
4. **KeyStoreListKeys(string passwd)**：
    - 打开密钥库。
    - 列出密钥库中的密钥信息。

FileManage 类功能流程

1. **SignAndEncryptMultiple(string passwd, string File, string[] all_Authorized_Uers_PublicKey, string userID)**：
   - 创建加密文件名。
   - 导出私钥。
   - 使用PGP库对文件进行签名和加密。
   - 删除临时私钥文件。
2. **SignAndEncryptSinge(string passwd, string File, string userID)**：
   - 类似于 `SignAndEncryptMultiple`，但只使用单个用户的公钥进行加密。
3. **SignAndEncryptString(string plainText, string passwd, string userID)**：
   - 导出私钥。
   - 使用PGP库对字符串进行签名和加密。
   - 删除临时私钥文件。
4. **Verify(string passwd, string File, string[] All_User_Public_key, string[] All_Users, string userID)**：
   - 尝试使用每个公钥解密并验证签名。
   - 如果签名验证成功，返回 `true`。
5. **DecryptAndVerify(string passwd, string File, string[] All_User_Public_key, string[] All_Users, string userID)**：
   - 解密文件并获取原始文件名。
   - 使用每个公钥验证签名。
   - 如果签名验证成功，打印成功信息。
6. **DecryptAndVerifyString(string signedAndEncryptedMessage, string passwd, string userID)**：
   - 导出私钥。
   - 使用PGP库解密并验证签名。
   - 删除临时私钥文件。
   - 打印解密后的字符串。

ClearTool 类功能流程

1. **ClearDelet(string path)**：
   - 判断路径是文件还是目录，调用相应的删除方法。
2. **ClearDeletDirectory(string dir)**：
   - 删除目录下的所有文件和子目录。
   - 重命名并删除当前目录。
3. **ClearDeletFile(string file)**：
   - 清空文件内容。
   - 重命名并删除文件。
4. **ClearFile(string file)**：
   - 打开文件，写入空内容清空文件。
   - 设置文件长度为0。

主程序 (Program 类)

1. Main(string[] args)：
    - 创建 `Global` 对象，展示文件夹创建。
    - 创建 `User` 对象，获取用户信息。
    - 创建 `KeyManage` 对象，生成密钥并导出。
    - 生成密码。
    - 进入模式选择界面，循环执行用户选择的操作模式。
    - 根据用户输入退出程序或返回用户界面。

具体代码和注释请参考附件

六、参考内容

[1] [GitHub - VivianYuan12138/BaseOpenPGP-FileManage: 基于公钥-私钥的思想，不依赖于本地Windows系统，设计了一个本地文件处理协议，基于open PGP实现本地加密文件夹。] (https://github.com/VivianYuan12138/BaseOpenPGP-FileManage?tab=readme-ov-file)

[2] [基于openPGP实现本地加密文件夹 | Yuanxinhang's Blog] (https://yuanxinhang.fun/2021/12/11/基于openPGP实现本地加密文件/)

[3] [Examples using DidiSoft OpenPGP products] (https://didisoft.com/examples/)

[4] [Kimi.ai - 帮你看更大的世界 (moonshot.cn)] (https://kimi.moonshot.cn/)

[5] [chatgpt] (https://chat.openai.com/)