

一、 环境说明

- 1. 开发环境: windows11 + visual studio + C++
- 2. 外部库

使用 IPworksOpenPGP

(1) 安装

OpenPGPC++	2024/4/17 19:26	文件夹	
IPWorksOpenPGP2022C++Edition_Tr...	2024/4/17 19:23	应用程序	4,042 KB

然后再 visual studio 内添加库和依赖项

常规

可执行文件目录	\$(VC_ExecutablePath_x64);\$(CommonExecutablePath)
包含目录	E:\IPworksPGP\OpenPGPC++\include;\$(IncludePath)
外部包含目录	\$(VC_IncludePath);\$(WindowsSDK_IncludePath);
引用目录	\$(VC_ReferencesPath_x64);
库目录	E:\IPworksPGP\OpenPGPC++\lib64;\$(LibraryPath)
Windows 运行库目录	\$(WindowsSDK_MetadataPath);
源目录	\$(VC_SourcePath);
排除目录	\$(CommonExcludePath);\$(VC_ExecutablePath_x64);\$(VC_L

公共项目内容

附加依赖项

ipworksopenpgp22.lib;%(AdditionalDependencies)

忽略所有默认库

忽略特定默认库

(2) 库内功能的使用

参考 IPworksOpenPGP 官方的参考文档

software | IPWorks OpenPGP 2022 C++ Edition

Sample Projects

Licensing Instructions

Class Reference

- CertMgr Class
- KeyMgr Class
- MIME Class
- OpenPGP Class
  - Introduction
  - Properties
  - Methods
  - Events
  - Types
  - Config Settings
  - Error Codes
- PFileMailer Class
- PHTMLMailer Class
- PIMAP Class

OpenPGP Class

[Properties](#) [Methods](#) [Events](#) [Config Settings](#) [Errors](#)

The OpenPGP class is used to encrypt/decrypt and sign/verify PGP messages.

Syntax

OpenPGP

Remarks

The OpenPGP class supports encrypting/decrypting and signing/verifying OpenPGP messages in the format specified by R

The `Encrypt`, `Sign`, and `SignAndEncrypt` methods are used to create a message to be sent to your partner. You can addition messages bound for multiple recipients with different keys, simultaneously encrypt and compress with the most popular cc algorithms, and control other aspects such as the encrypting algorithm to use.

When a message is received, the `Decrypt`, `VerifySignature`, and `DecryptAndVerifySignature` methods are used to process th

- 3. 使用环境: 程序可在 windows 环境下执行

## 二、程序设计

### 1、类构造说明

(1) MyKeyMgr 类负责管理用户名和其对应的公钥和私钥

```
// 自定义的KeyMgr类, 继承自KeyMgr类
class MyKeyMgr : public KeyMgr
{
private:
    OpenPGP* mypgp;
public:
    vector<char*> allusers;//记录所有拥有密钥用户名
    char keyringDir[LINE_LEN]; // 密钥环的目录
    char myusername[LINE_LEN]; //当前用户名

    virtual int FireKeyList(KeyMgrKeyListEventParams* e)
    {
        //cout << e->UserId << " ";
        char* temp = new char(LINE_LEN);
        strcpy(temp, e->UserId);
        allusers.push_back(temp);//添加到用户记录中
        return 0;
    }

    void SetKeyringDir();//设定密钥环目录路径
    int CheckUserExistence(char keyusername[]);//检查用户是否存在密钥
    void CreateUserKey(char keyusername[]);//为用户创建密钥
}
```

(2) MyFileMgr 负责管理利用 OpenPGP 对文件进行加密解密等操作

```
class MyFileMgr //管理文件操作
{
    OpenPGP* mypgp;
    MyKeyMgr* mykeymgr;

public:
    void FileMenu();//显示文件菜单
    void FileOperation(int opt);//进行文件操作
    void SetFileName(char inputfile[], char outputfile[]);

    MyFileMgr(OpenPGP* inmypgp, MyKeyMgr* inmykeymgr) {
        mypgp = inmypgp;
        mykeymgr = inmykeymgr;
    }
};
```

## 2、程序结构设计

```
int main()
{
    char username[LINE_LEN]; //用户名
    GetWinUserName(username); //获取当前windows账户用户名
    cout << "欢迎用户 " << username << " 使用本文件加密程序" << endl;

    OpenPGP pgp; //IPWorks OpenPGP组件
    MyKeyMgr keymgr(username, &pgp); //密钥管理

    MyFileMgr filemgr(&pgp, &keymgr); //文件操作管理
    filemgr.FileMenu(); //显示文件菜单

    return 0;
}
```

- (1) 启动程序后，利用 windows 库实现获取当前用户名
- (2) 构造相关类并初始化。
- (3) 在初始化 MyKeyMgr 类的时候，在密钥环目录里检测是否有当前用户对应的密钥，若不存在，则创建。

```
void MyFileMgr::FileMenu() //显示文件菜单
{
    while (1) {
        system("cls");
        cout << "加密文件-----按 1" << endl;
        cout << "解密文件-----按 2" << endl;
        cout << "签名文件-----按 3" << endl;
        cout << "验证文件-----按 4" << endl;
        cout << "签名并加密文件-----按 5" << endl;
        cout << "解密并验证文件-----按 6" << endl;
        cout << "退出程序-----按 0" << endl;
    }
}
```

- (4) 然后显示执行不同功能的菜单。按下不同的按键执行不同的程序。
- (5) 所有功能都需要一个输入文件和一个输出文件。在加密过程中，加密者用户名自动设置为当前用户，自行设置接收者的用户名。（若密钥环不存在接收者的密钥，则创建一个。）
- (6) 在解密时，利用当前用户名的私钥尝试解密；若不对应，则出错。
- (7) 在签名时，签名者用户名自动设置为当前用户。
- (8) 在验证文件时，验证者需要输入可能的签名者的用户名；若不对应，则出错。
- (9) 在执行对应的功能时，利用库函数设置密钥环路径和相关操作的用户名。如果发生错误，则输出对应的错误类型，停止执行程序。

### 三、 执行演示

#### 1、准备工作

##### (1) 可执行程序

keyring

testfile

ipworksopenpgp22.dll

OpenPGPproj.exe

##### (2) 相关文件夹

- a、在当前目录下创建 keyring 文件夹，用于储存密钥。
- b、创建 testfile 文件夹，用于储存明文密文等测试用例。

##### (3) 在 windows 创建两个账户



#### 2、执行程序

##### (1) 读取应户名。设置密钥环路径

```
欢迎用户 SiShengyu 使用本文件加密程序
请输入密钥环存放的目录路径
(若直接回车，默认为当前目录下的keyring文件夹)：
```

```
加密文件-----按 1
解密文件-----按 2
签名文件-----按 3
验证文件-----按 4
签名并加密文件-----按 5
解密并验证文件-----按 6
退出程序-----按 0
```


##### (2) 加密功能


```
当前用户: SiShengyu
请设置操作的输入文件路径: testfile/orifile1.txt
请设置操作的输出文件路径: testfile/forsisus.txt
是否覆盖原文件? (y/n) y
请输入允许调阅该文件的用户名: sisus
文件加密完成
请按任意键继续 . . . |
```

首先加密 orifile1.txt, 接受者为 sisus

```
当前用户: SiShengyu
请设置操作的输入文件路径: testfile/orifile1.txt
请设置操作的输出文件路径: testfile/forapple.txt
是否覆盖原文件? (y/n) y
请输入允许调阅该文件的用户名: apple
文件加密完成
请按任意键继续 . . . |
```

再次加密 orifile1.txt, 接受者为 apple

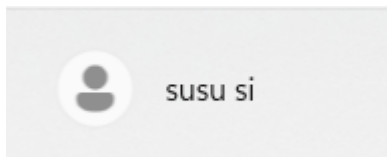
 forapple.txt

 forsisus.txt

 orifile1.txt

### (3) 解密功能

切换到 sisus 用户

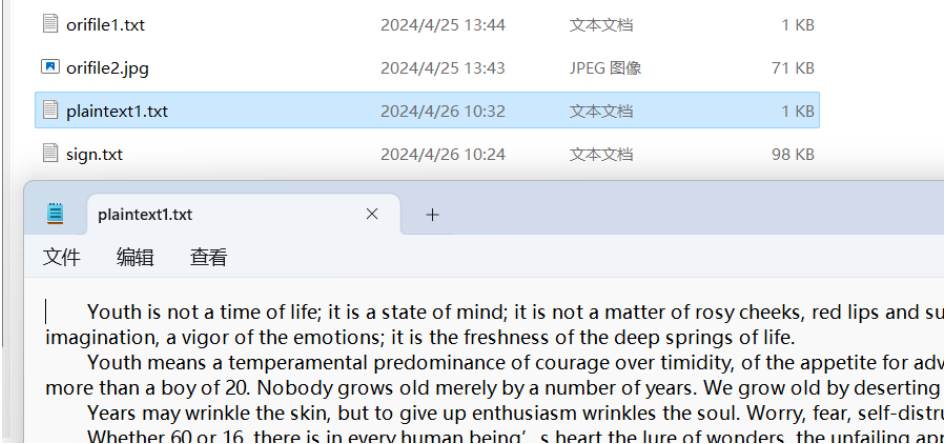


```
欢迎用户 sisus 使用本文件加密程序
请输入密钥环存放的目录路径
(若直接回车, 默认为当前目录下的keyring文件夹): |
```

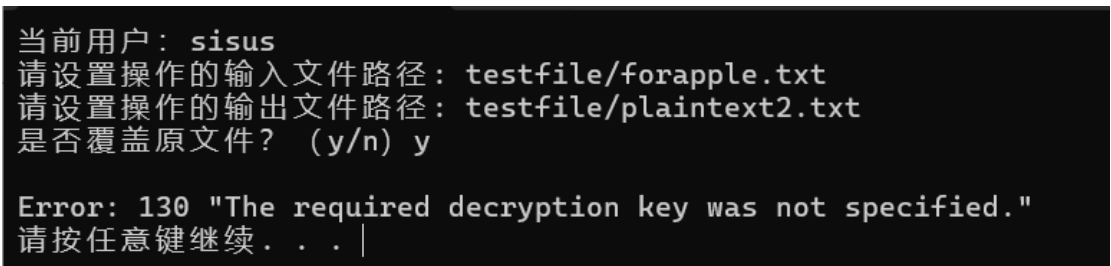
a. 尝试解密 forsisus.txt

```
C:\Windows\System32\cmd.e  ×  +  v
当前用户: sisus
请设置操作的输入文件路径: testfile/forsisus.txt
请设置操作的输出文件路径: testfile/plaintext1.txt
是否覆盖原文件? (y/n) y
文件解密完成
请按任意键继续 . . . |
```

解密成功

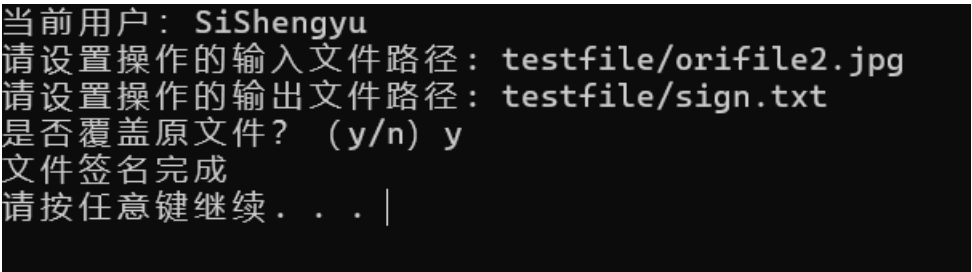


b. 尝试解密 forapple.txt

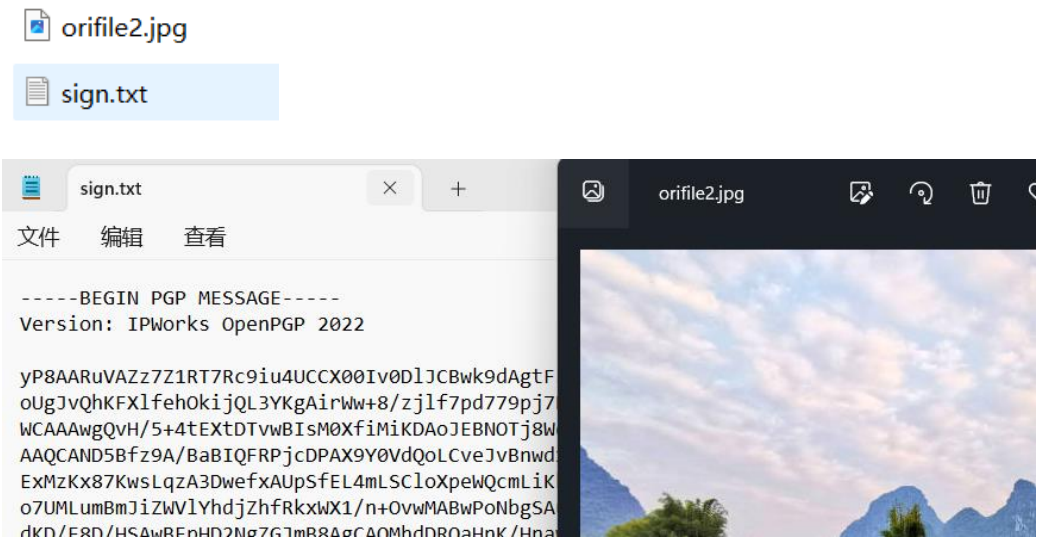


解密失败，因为当前用户并不是 apple。报错。

(4) 签名功能



利用图片，生成签名



### (5) 验证功能

- a. 首先尝试验证 sign.txt，判断是否是 SiShengyu 的签名

```
C:\Windows\System32\cmd.e  X  +  v

当前用户: sisus
请设置操作的输入文件路径: testfile/sign.txt
请设置操作的输出文件路径: testfile/verify.jpg
是否覆盖原文件? (y/n) y
请输入准备验证的文件发布者的用户名: SiShengyu
文件验证完成
请按任意键继续. . . |
```

验证成功

- b. 尝试更改 sign.txt 后再验证。

<pre>-----BEGIN PGP MESSAGE----- Version: IPWorks OpenPGP 2022  yP8AARuVAZz7Z1RT7Rc9iu4U oUgJvQhKFXIfehOkijQL3YKgA</pre>	<pre>-----BEGIN PGP MESSAGE----- Version: IPWorks OpenPGP 2022  P8AARuVAZz7Z1RT7Rc9iu4UCCX oUgJvQhKFXIfehOkijQL3YKgAirV WCAAAwgcQvH/5+4tFXtDTvwRlcl</pre>
--	---

签名有误，报错

```
C:\Windows\System32\cmd.e  X  +  v

当前用户: sisus
请设置操作的输入文件路径: testfile/sign.txt
请设置操作的输出文件路径: testfile/v2.jpg
是否覆盖原文件? (y/n) y
请输入准备验证的文件发布者的用户名: SiShengyu

Error: 103 "Checksum failed."
请按任意键继续. . . |
```

- c. 尝试验证是否为其他人的签名

签名者用户名有误，报错

```
C:\Windows\System32\cmd.e  X  +  v

当前用户: sisus
请设置操作的输入文件路径: testfile/sign.txt
请设置操作的输出文件路径: testfile/v2.txt
是否覆盖原文件? (y/n) y
请输入准备验证的文件发布者的用户名: apple

Error: 114 "The required signer key could not be found."
请按任意键继续. . . |
```

### (6) 剩余功能执行效果类似