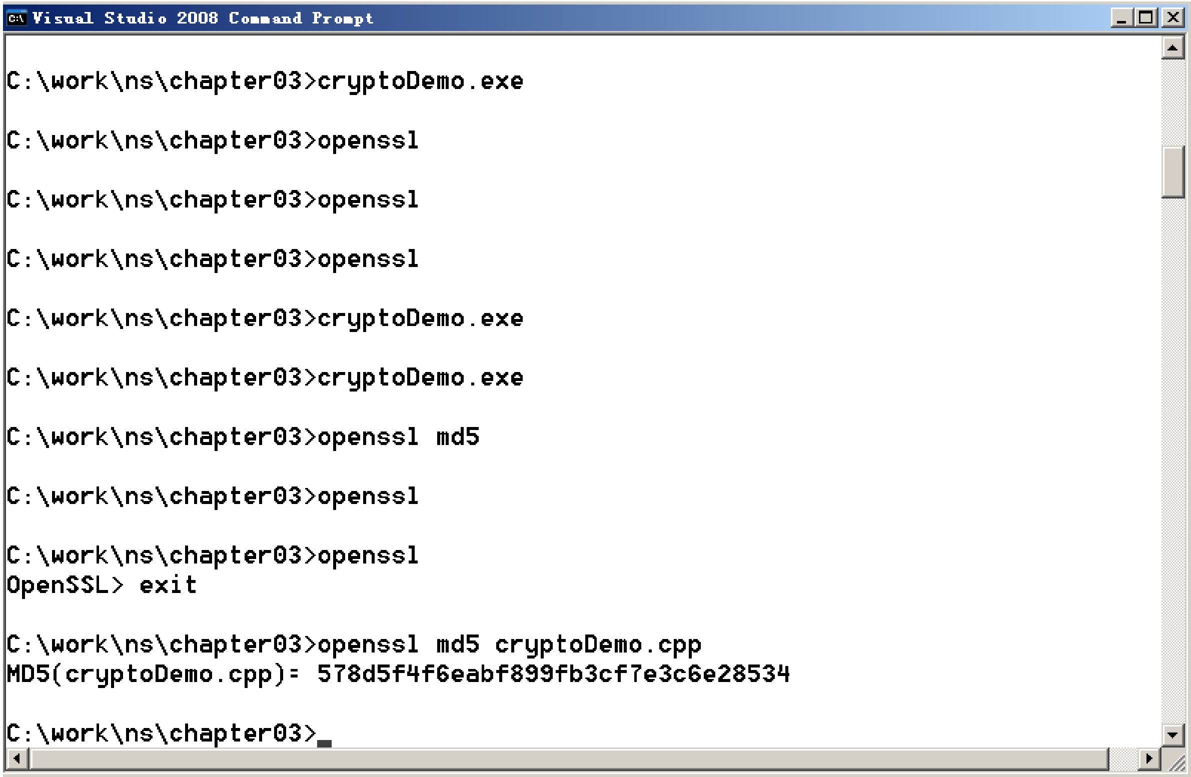
信息安全导论第二次实验

PB19051183 吴承泽

2.3.1 使用 OpenSSL 的常用命令

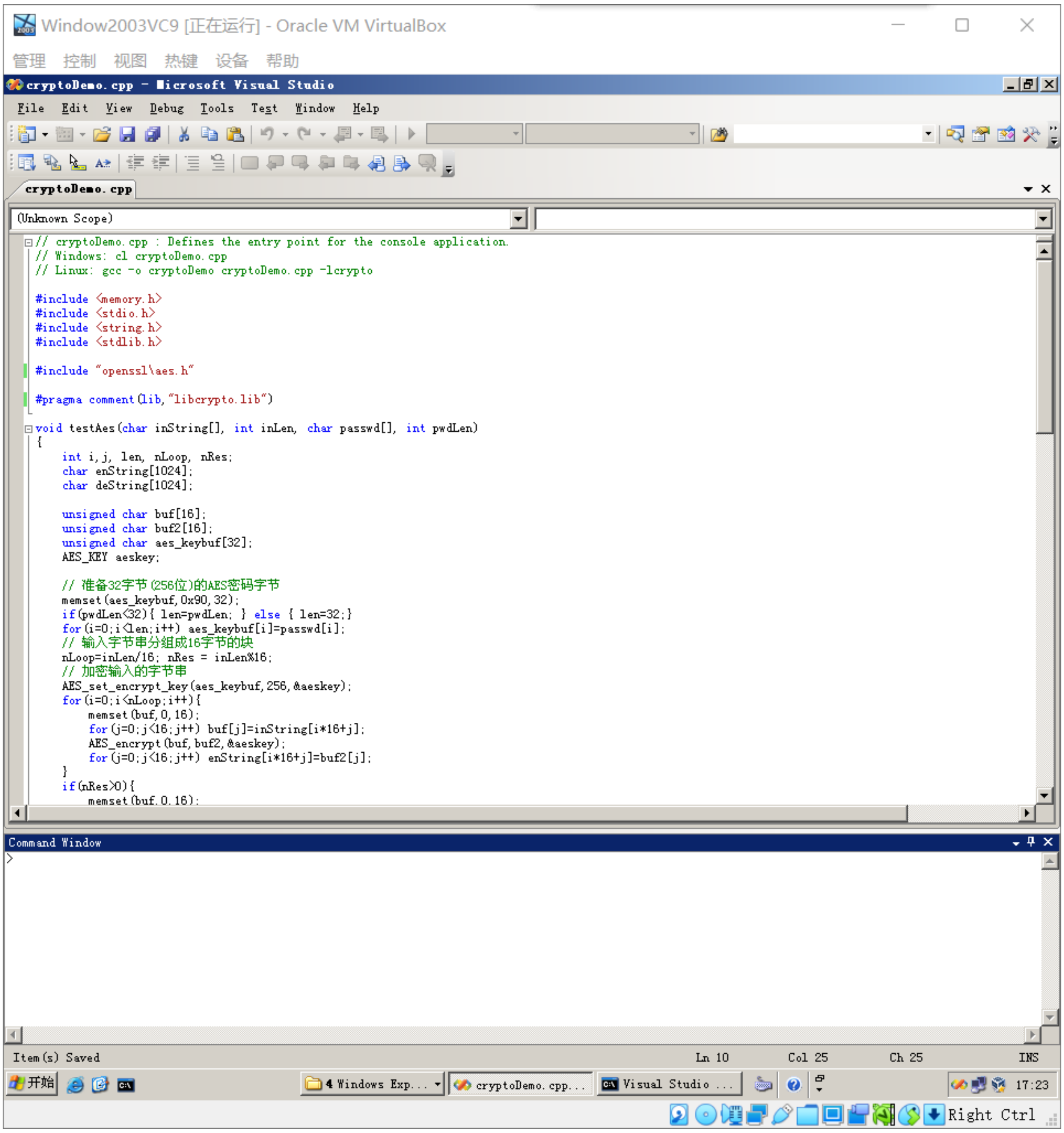
安装好Win32OpenSSL文件后，使用OpenSSL命令，通过执行命令 `openssl md5 cryptoDemo.cpp`，产生的串如下所示：



```
C:\work\ns\chapter03>cryptoDemo.exe

C:\work\ns\chapter03>openssl

C:\work\ns\chapter03>openssl

C:\work\ns\chapter03>openssl

C:\work\ns\chapter03>cryptoDemo.exe

C:\work\ns\chapter03>cryptoDemo.exe

C:\work\ns\chapter03>openssl md5

C:\work\ns\chapter03>openssl

C:\work\ns\chapter03>openssl
OpenSSL> exit

C:\work\ns\chapter03>openssl md5 cryptoDemo.cpp
MD5(cryptoDemo.cpp)= 578d5f4f6eabf899fb3cf7e3c6e28534

C:\work\ns\chapter03>_
```

2.3.2 利用 OpenSSL 编程实现 AES 的加密、解密

修改cryptoDemo.cpp如下所示：

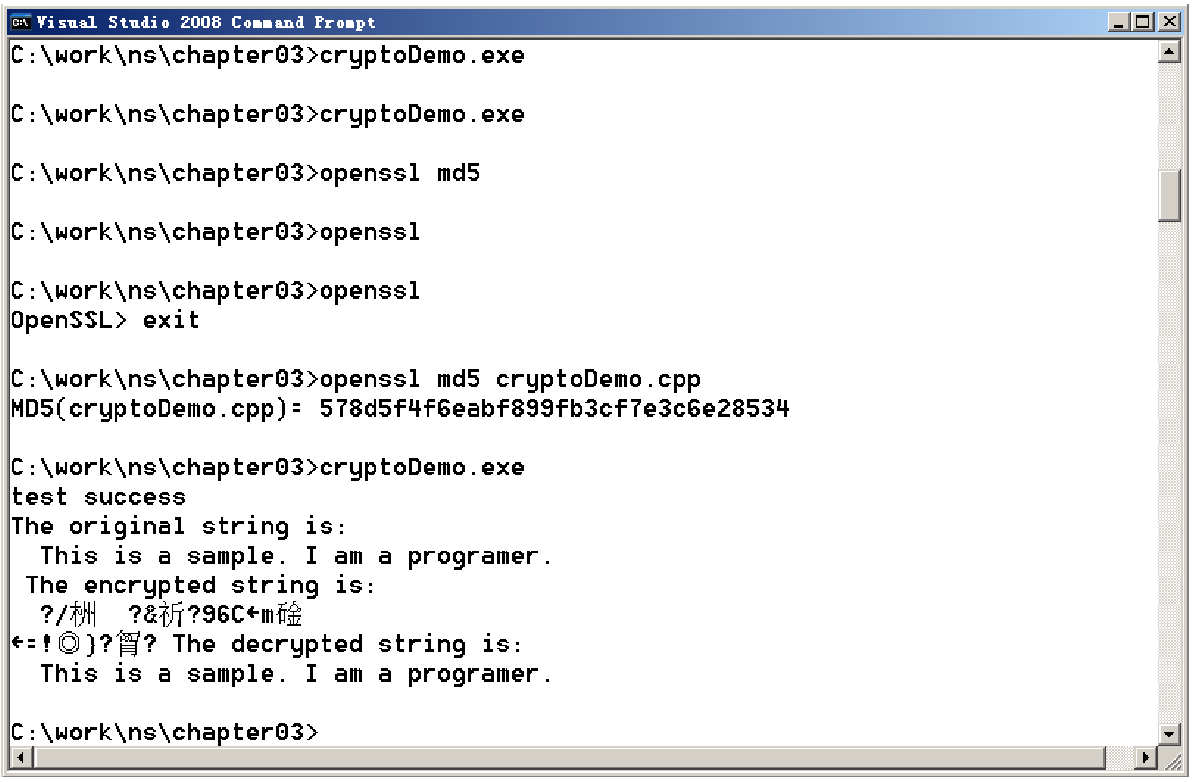

Window2003VC9 [正在运行] - Oracle VM VirtualBox
管理 控制 视图 热键 设备 帮助
cryptoDemo.cpp - Microsoft Visual Studio
File Edit View Debug Tools Test Window Help
cryptoDemo.cpp
(Unknown Scope)
Command Window
Item(s) Saved
Ln 10
Col 25
Ch 25
INS
开始
4 Windows Exp...
cryptoDemo.cpp...
Visual Studio ...
17:23
Right Ctrl

```
// cryptoDemo.cpp : Defines the entry point for the console application.
// Windows: cl cryptoDemo.cpp
// Linux: gcc -o cryptoDemo cryptoDemo.cpp -lcrypto

#include <memory.h>
#include <stdio.h>
#include <string.h>
#include <stdlib.h>

#include "openssl\aes.h"

#pragma comment(lib,"libcrypto.lib")

void testAes(char inString[], int inLen, char passwd[], int pwdLen)
{
    int i,j, len, nLoop, nRes;
    char enString[1024];
    char deString[1024];

    unsigned char buf[16];
    unsigned char buf2[16];
    unsigned char aes_keybuf[32];
    AES_KEY aeskey;

    // 准备32字节(256位)的AES密码字节
    memset(aes_keybuf,0x90,32);
    if(pwdLen<32){ len=pwdLen; } else { len=32;}
    for(i=0;i<len;i++) aes_keybuf[i]=passwd[i];
    // 输入字节串分组成16字节的块
    nLoop=inLen/16; nRes = inLen%16;
    // 加密输入的字节串
    AES_set_encrypt_key(aes_keybuf,256,&aeskey);
    for(i=0;i<nLoop;i++){
        memset(buf,0,16);
        for(j=0;j<16;j++) buf[j]=inString[i*16+j];
        AES_encrypt(buf,buf2,&aeskey);
        for(j=0;j<16;j++) enString[i*16+j]=buf2[j];
    }
    if(nRes>0){
        memset(buf,0,16);
```

通过指令编译得到如下输出：

```
C:\work\ns\chapter03>cl cryptoDemo.cpp
Microsoft (R) 32-bit C/C++ Optimizing Compiler Version 15.00.21022.08 for 80x86
Copyright (C) Microsoft Corporation.  All rights reserved.

cryptoDemo.cpp
Microsoft (R) Incremental Linker Version 9.00.21022.08
Copyright (C) Microsoft Corporation.  All rights reserved.

/out:cryptoDemo.exe
cryptoDemo.obj
```

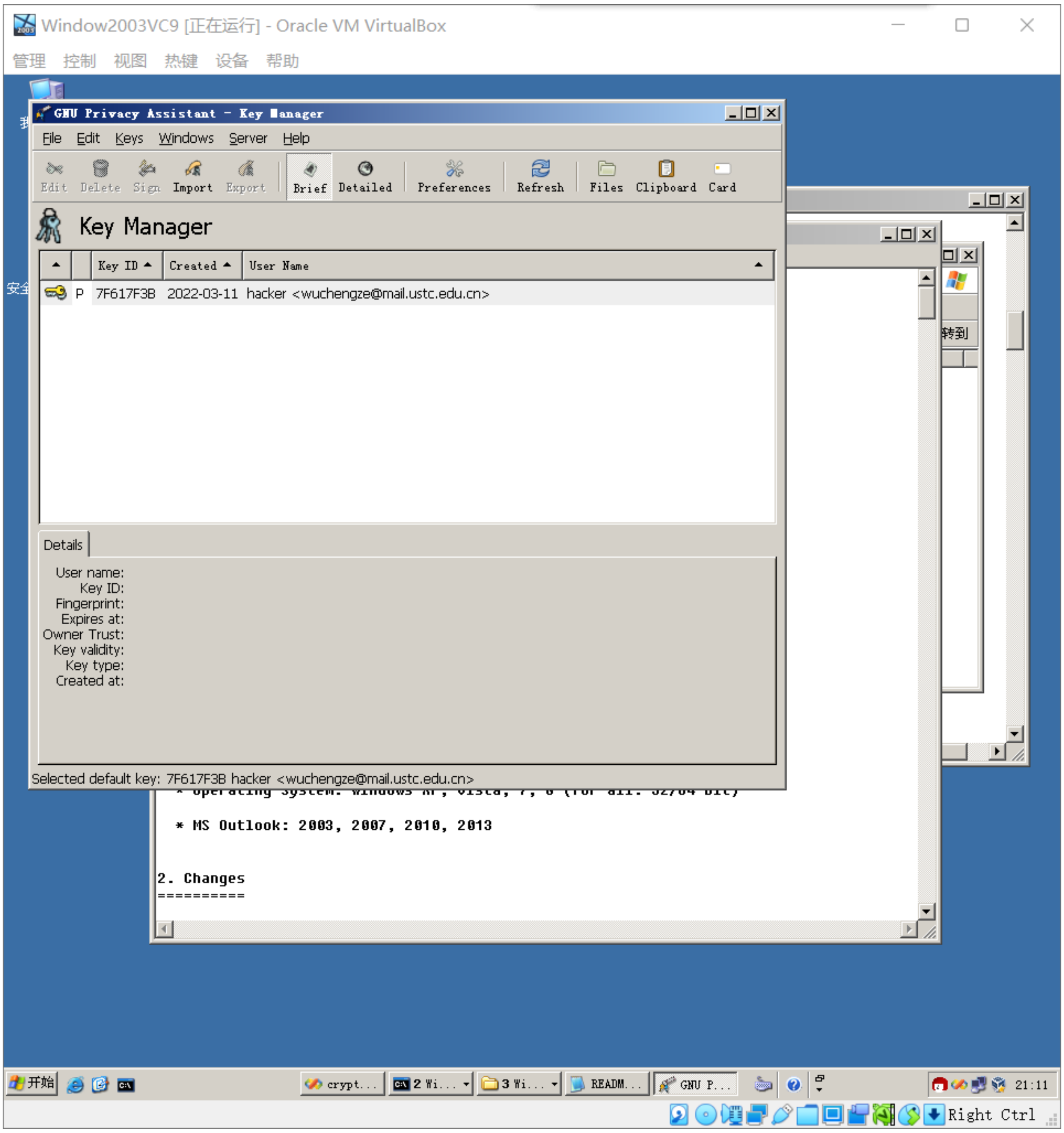
在运行得到如下结果：



```
C:\work\ns\chapter03>cryptoDemo.exe

C:\work\ns\chapter03>cryptoDemo.exe

C:\work\ns\chapter03>openssl md5

C:\work\ns\chapter03>openssl

C:\work\ns\chapter03>openssl
OpenSSL> exit

C:\work\ns\chapter03>openssl md5 cryptoDemo.cpp
MD5(cryptoDemo.cpp)= 578d5f4f6eabf899fb3cf7e3c6e28534

C:\work\ns\chapter03>cryptoDemo.exe
test success
The original string is:
  This is a sample. I am a programer.
 The encrypted string is:
  ?/梆  ?&祈?96C←m硂
←=!◎}?胥? The decrypted string is:
  This is a sample. I am a programer.

C:\work\ns\chapter03>
```

2.3.3 用 PGP 实现加密和解密

下载完Gpg4win2.2.3后，打开GPA，以用户名hacker，电子邮件地址wuchengze@mail.ustc.edu.cn产生一对密钥。



Window2003VC9 [正在运行] - Oracle VM VirtualBox
管理 控制 视图 热键 设备 帮助
GNU Privacy Assistant - Key Manager
File Edit Keys Windows Server Help
Edit Delete Sign Import Export Brief Detailed Preferences Refresh Files Clipboard Card
Key Manager
Key ID Created User Name
Details
开始
crypt... 2 Wi... 3 Wi... READM... GNU P...
21:11
Right Ctrl

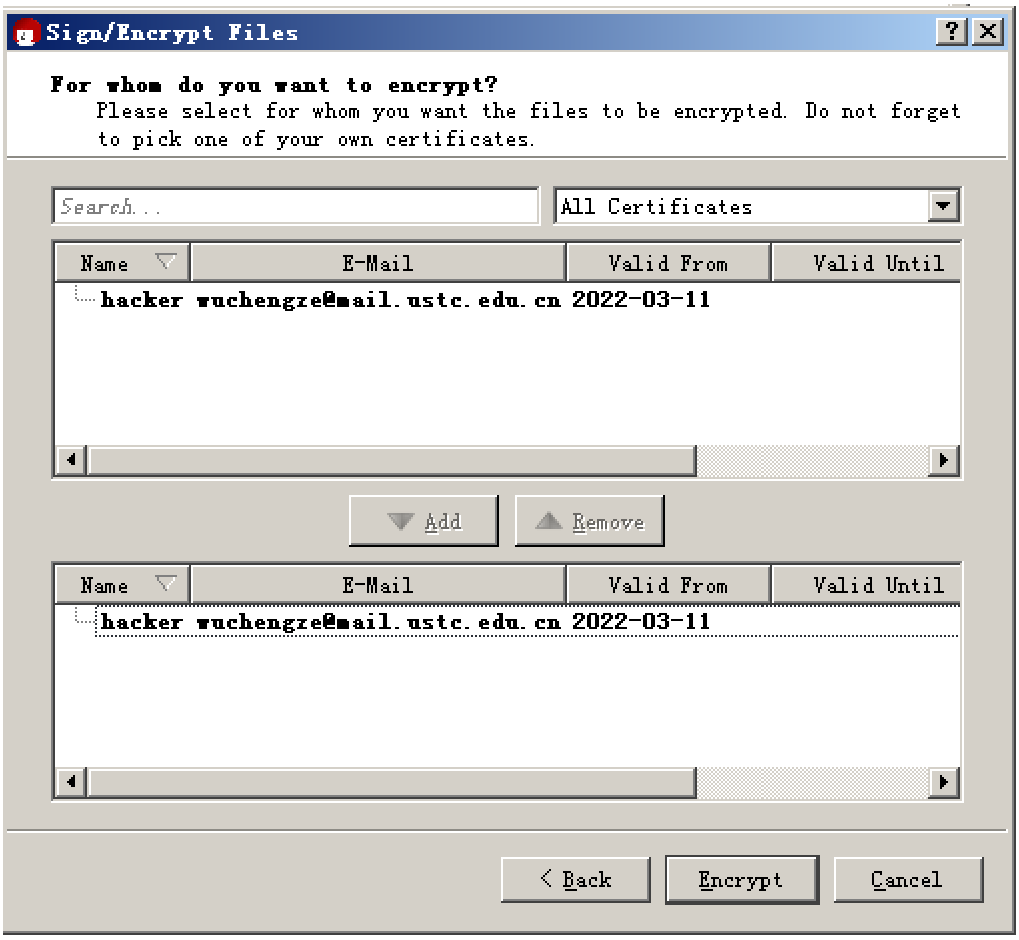
	Key ID	Created	User Name
P	7F617F3B	2022-03-11	hacker <wuchengze@mail.ustc.edu.cn>

User name:
Key ID:
Fingerprint:
Expires at:
Owner Trust:
Key validity:
Key type:
Created at:

Selected default key: 7F617F3B hacker <wuchengze@mail.ustc.edu.cn>

```
 * MS Outlook: 2003, 2007, 2010, 2013

2. Changes
==========
```

导出公钥后，文件选择cryptoDemo.cpp，右键选择encrypt and sign



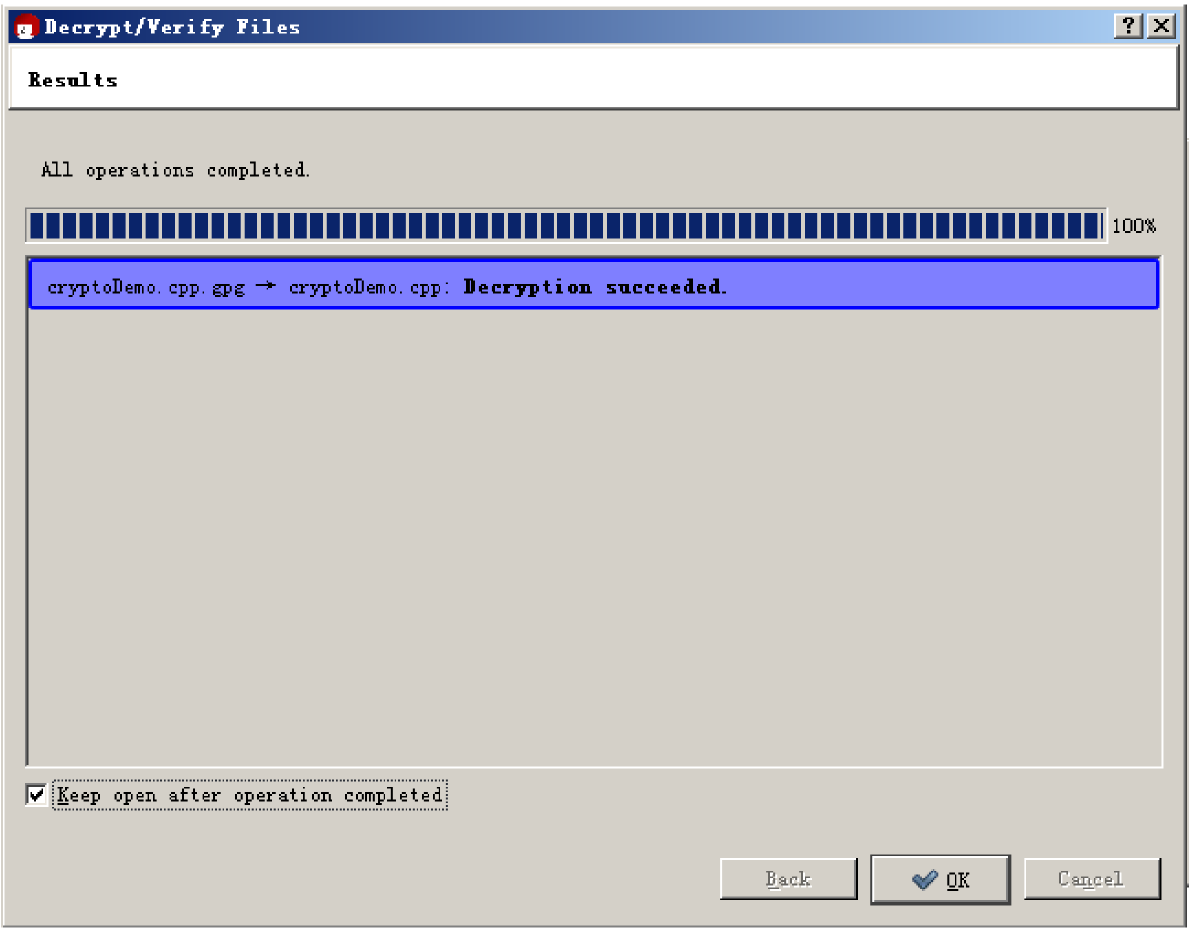
Sign/Encrypt Files

For whom do you want to encrypt?
Please select for whom you want the files to be encrypted. Do not forget to pick one of your own certificates.

Name	E-Mail	Valid From	Valid Until
hacker	wuchengze@mail.ustc.edu.cn	2022-03-11	

Name	E-Mail	Valid From	Valid Until
hacker	wuchengze@mail.ustc.edu.cn	2022-03-11	

成功导出后得到cryptoDemo.cpp.gpg加密文件，通过右键选择decrypt，得到与源文件相同的文件，解密成功。



Decrypt/Verify Files

Results

All operations completed.

100%

cryptoDemo.cpp.gpg → cryptoDemo.cpp: **Decryption succeeded.**

☑ Keep open after operation completed

2.5 做实验并写实验报告

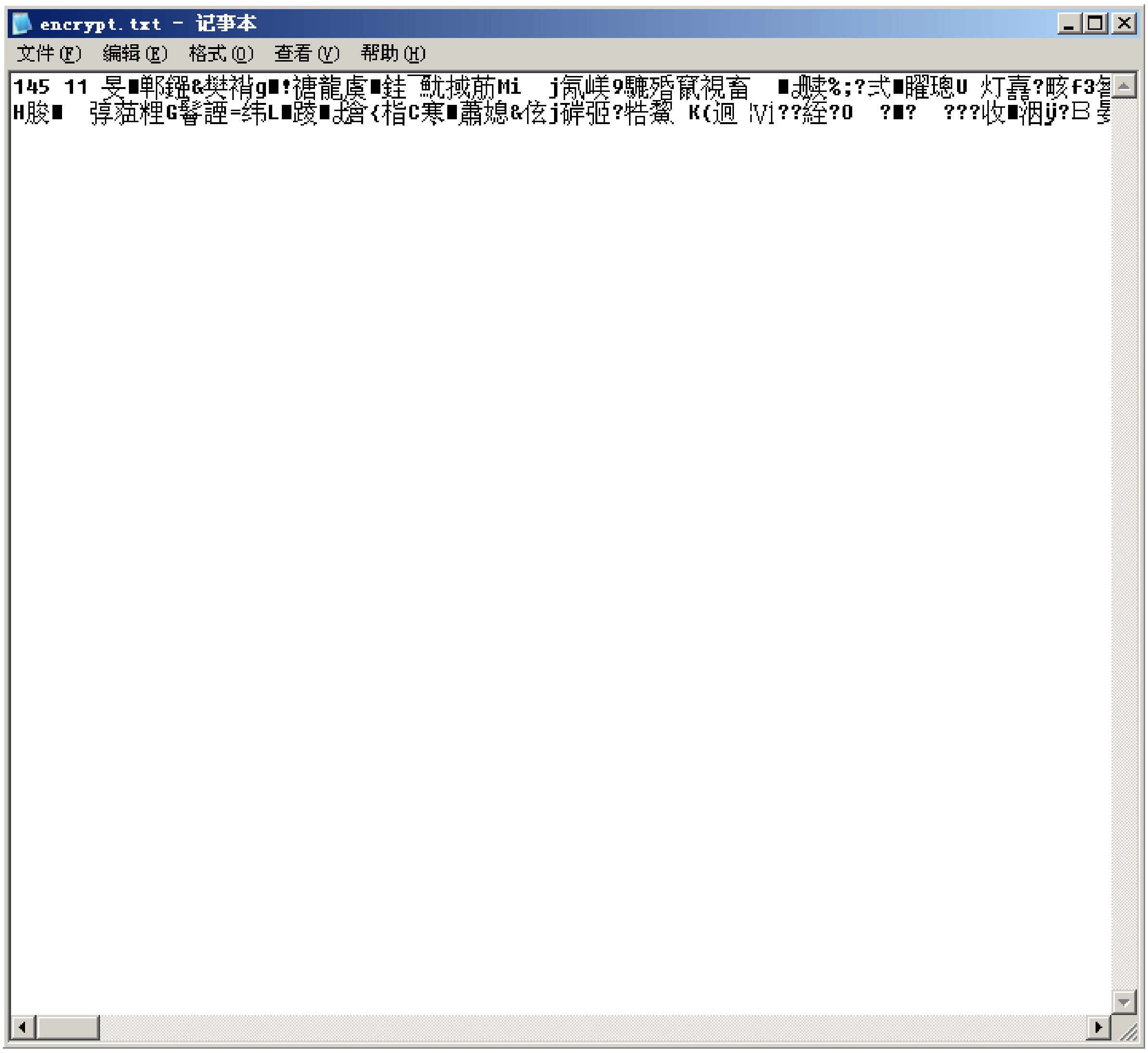
修改例程encfile.cpp如下:

```
#include <memory.h>
#include <stdio.h>
#include <string.h>
#include <stdlib.h>

#include "openssl\aes.h"

int nLoop;
int nRes;

#pragma comment(lib,"libcrypto.lib")

void encrypt(char inString[], int inLen, char passwd[], int pwdLen)
{
    int i,j, len;
    char enString[35536];
    nLoop = inLen/16;
    nRes = inLen%16;
    unsigned char buf[16];
    unsigned char buf2[16];
    unsigned char aes_keybuf[32];
    AES_KEY aeskey;

    // 准备32字节(256位)的AES密码字节
    memset(aes_keybuf,0x90,32);
    if(pwdLen<32){ len=pwdLen; } else { len=32;}
    for(i=0;i<len;i++) aes_keybuf[i]=passwd[i];
    // 输入字节串分组成16字节的块
    nLoop=inLen/16; nRes = inLen%16;
    // 加密输入的字节串
    AES_set_encrypt_key(aes_keybuf,256,&aeskey);
    for(i=0;i<nLoop;i++){
        memset(buf,0,16);
        for(j=0;j<16;j++) buf[j]=inString[i*16+j];
        AES_encrypt(buf,buf2,&aeskey);
        for(j=0;j<16;j++) enString[i*16+j]=buf2[j];
    }
    if(nRes>0){
        memset(buf,0,16);
        for(j=0;j<nRes;j++) buf[j]=inString[i*16+j];
        AES_encrypt(buf,buf2,&aeskey);
        for(j=0;j<16;j++) enString[i*16+j]=buf2[j];
        //puts("encrypt");
    }
    enString[i*16+j]=0;

    FILE *fp;
    if((fp = fopen("encrypt.txt", "wb")) == NULL)
        exit(-1);
    fclose(fp);
    if((fp = fopen("encrypt.txt", "ab")) == NULL)
        exit(-1);

    fprintf(fp,"%d %d ", nLoop, nRes);
    int k;
    for(k = 0; k <= i*16 +j;k++)
        //fprintf(fp,"%c",enString[k]);
        fputc(enString[k], fp);

    fclose(fp);


}

void decrypt(char enString[], int enLen, char passwd[], int pwdLen)
{
    int i,j, len;
    char deString[35536];

    unsigned char buf[16];
    unsigned char buf2[16];
    unsigned char aes_keybuf[32];
    AES_KEY aeskey;
    // 密文串的解密
    memset(aes_keybuf,0x90,32);
    if(pwdLen<32){ len=pwdLen; } else { len=32;}
    for(i=0;i<len;i++) aes_keybuf[i]=passwd[i];
    AES_set_decrypt_key(aes_keybuf,256,&aeskey);
    for(i=0;i<nLoop;i++){
        memset(buf,0,16);
        for(j=0;j<16;j++) buf[j]=enString[i*16+j];
        AES_decrypt(buf,buf2,&aeskey);
        for(j=0;j<16;j++) deString[i*16+j]=buf2[j];
    }
    if(nRes>0){
        memset(buf,0,16);
        for(j=0;j<16;j++) buf[j]=enString[i*16+j];
        AES_decrypt(buf,buf2,&aeskey);
        for(j=0;j<16;j++) deString[i*16+j]=buf2[j];
        //puts("decrypt");
    }
    deString[i*16+nRes]=0;
    FILE *fp;
    if((fp = fopen("decrypt.cpp", "wb")) == NULL)
        exit(-1);

    fprintf(fp,"%s", deString);
    fclose(fp);

}

int main(int argc, char* argv[])
{
    FILE *fp;
    char str[35536];
    char pwd[128];
    if(strcmp(argv[1], "enc") == 0)
    {
        if((fp = fopen(argv[2], "rb")) == NULL)
            exit(-1);
        char ch = fgetc(fp);
        int i = 0;
        while(ch != EOF)
        {
            str[i] = ch;
            i++;
            //fscanf(fp,"%c",ch);
            ch = fgetc(fp);
        }
        str[i] = 0;
        encrypt(str, strlen(str), argv[3], strlen(argv[3]));
        fclose(fp);
    }
    else if(strcmp(argv[1], "dec") == 0)
    {
        if((fp = fopen("encrypt.txt", "rb")) == NULL)
            exit(-1);

        fscanf(fp,"%d %d ",&nLoop, &nRes);
        char ch = fgetc(fp);
        int i = 0;
        int length = (nRes)? 16 : 0;
        length += 16 * nLoop;
        while(i < length)
        {
            str[i] = ch;
            //printf("%c",ch);
            i++;
            ch = fgetc(fp);
        }
        str[length] = 0;
        decrypt(str, length, argv[3], strlen(argv[3]));
        fclose(fp);
    }
    else
    {
        printf("error");
        exit(-1);
    }
    return 0;
}
```

在VS2008终端中使用指令行编译该文件，以PB19051183为密码执行加密解密：

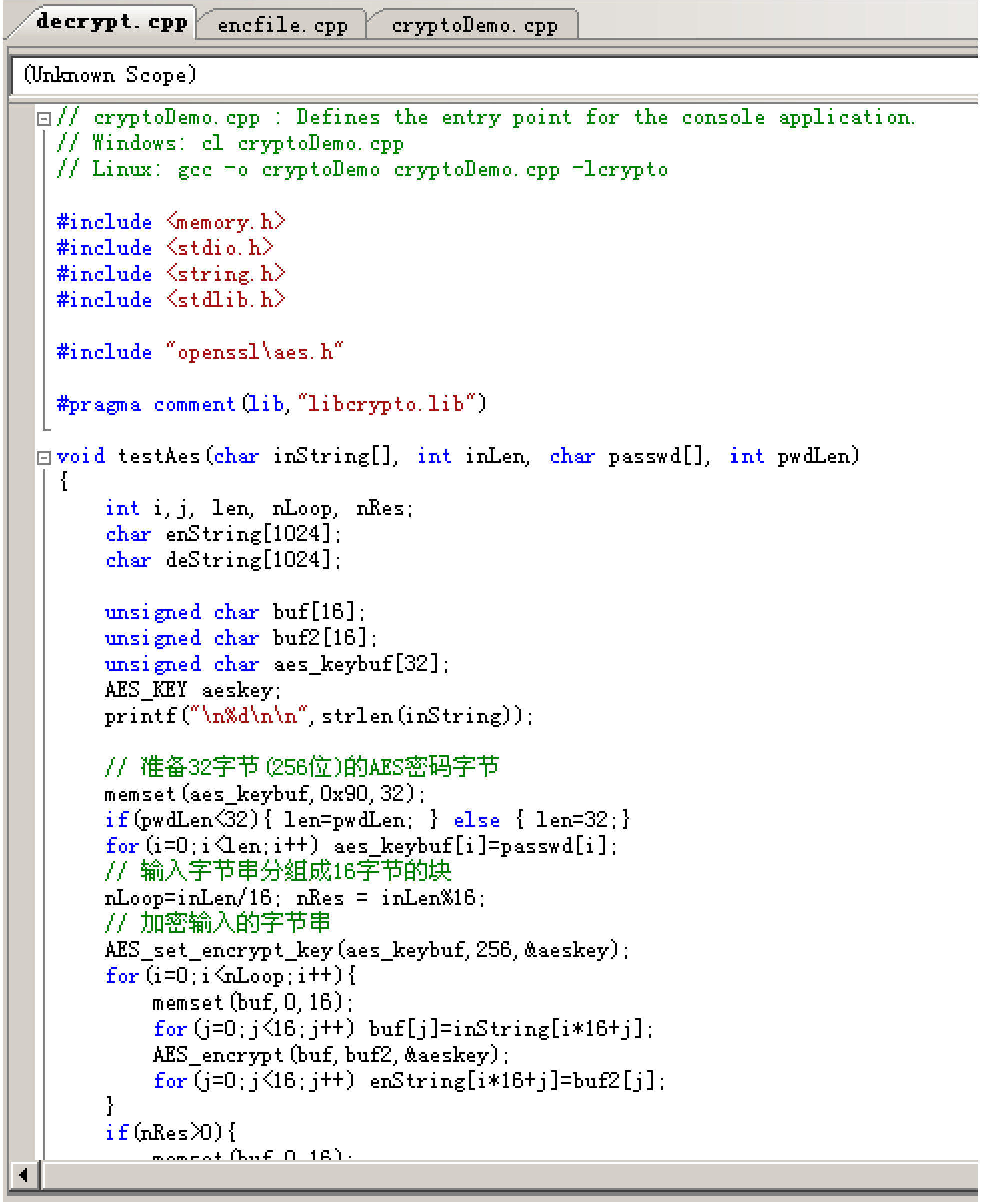
```
cl encfile.cpp
encfile.exe enc cryptoDemo.cpp PB19051183
encfile.exe dec encrypt.txt PB19051183
```

得到的加密文件如下所示：



```
145 11 旻■郸鏹&樊褙g■!禟龍虞■鞋 魷摵莇Mi   j氝崨9驋瘖竄視畜   ■虪%;?式■矔璁U 灯嘉?晐f3
H腠■   彉菹粴G謦謰=纬L■踜■擿{指C寒■蕭媳&佉j碑弫?牿鸏 K(逥 |Vj??絟?0   ?■?   ???收■衵ÿ?B 旻
```

经解密后的文件decrypt.cpp如下所示，与cryptoDemo.cpp完全一致，可以验证该程序的正确性。



```
// cryptoDemo.cpp : Defines the entry point for the console application.
// Windows: cl cryptoDemo.cpp
// Linux: gcc -o cryptoDemo cryptoDemo.cpp -lcrypto

#include <memory.h>
#include <stdio.h>
#include <string.h>
#include <stdlib.h>

#include "openssl\aes.h"

#pragma comment(lib,"libcrypto.lib")

void testAes(char inString[], int inLen, char passwd[], int pwdLen)
{
    int i,j, len, nLoop, nRes;
    char enString[1024];
    char deString[1024];

    unsigned char buf[16];
    unsigned char buf2[16];
    unsigned char aes_keybuf[32];
    AES_KEY aeskey;
    printf("\n%d\n\n",strlen(inString));

    // 准备32字节(256位)的AES密码字节
    memset(aes_keybuf,0x90,32);
    if(pwdLen<32){ len=pwdLen; } else { len=32;}
    for(i=0;i<len;i++) aes_keybuf[i]=passwd[i];
    // 输入字节串分組成16字节的块
    nLoop=inLen/16; nRes = inLen%16;
    // 加密输入的字节串
    AES_set_encrypt_key(aes_keybuf,256,&aeskey);
    for(i=0;i<nLoop;i++){
        memset(buf,0,16);
        for(j=0;j<16;j++) buf[j]=inString[i*16+j];
        AES_encrypt(buf,buf2,&aeskey);
        for(j=0;j<16;j++) enString[i*16+j]=buf2[j];
    }
    if(nRes>0){
        memset(buf,0,16);
```