信息安全导论 Lab2

廖洲洲 PB17081504

一、 实验目的

1. 掌握 OpenSSL 的命令
2. 掌握在 C 程序中使用 OpenSSL 的方法
3. 掌握 PGP 的使用

二、 实验内容

1. 使用 OpenSSL 的常用命令
2. 利用 OpenSSL 编程实现 RSA 加密、解密
3. 用 PGP 实现加密和解密

三、 实验步骤

1. 使用 OpenSSL 的常用命令

1.1在 Ubuntu Linux 系统中运行以下命令安装 openssl 库：

```
liaozz@liaozz-VirtualBox:~$ sudo apt-get install libssl-dev
[sudo] password for liaozz:
对不起，请重试。
[sudo] password for liaozz:
对不起，请重试。
[sudo] password for liaozz:
正在读取软件包列表... 完成
正在分析软件包的依赖关系树
正在读取状态信息... 完成
将会安装下列额外的软件包：
  libssl-doc zlib1g-dev
下列【新】软件包将被安装：
  libssl-dev libssl-doc zlib1g-dev
升级了 0 个软件包，新安装了 3 个软件包，要卸载 0 个软件包，有 61 个软件包未被升
级。
需要下载 2,132 kB 的软件包。
解压缩后会消耗掉 6,908 kB 的额外空间。
您希望继续执行吗？ [Y/n] y
获取：1 http://cn.archive.ubuntu.com/ubuntu/ trusty-updates/main zlib1g-dev i386
 1:1.2.8.dfsg-1ubuntu1.1 [165 kB]
获取：2 http://cn.archive.ubuntu.com/ubuntu/ trusty-updates/main libssl-dev i386
 1.0.1f-1ubuntu2.27 [995 kB]
获取：3 http://cn.archive.ubuntu.com/ubuntu/ trusty-updates/main libssl-doc all
1.0.1f-1ubuntu2.27 [972 kB]
下载 2,132 kB，耗时 23秒 (92.
正在选中未选择的软件包 zlib1g
(正在读取数据库 ... 系统当前共安装有 169609 个文件和目录。)
正准备解包 .../zlib1g-dev_1%3a1.2.8.dfsg-1ubuntu1.1_i386.deb  ...
正在解包 zlib1g-dev:i386 (1:1.2.8.dfsg-1ubuntu1.1) ...
正在选中未选择的软件包 libssl-dev:i386。
正准备解包 .../libssl-dev_1.0.1f-1ubuntu2.27_i386.deb  ...
正在解包 libssl-dev:i386 (1.0.1f-1ubuntu2.27) ...
正在选中未选择的软件包 libssl-doc。
正准备解包 .../libssl-doc_1.0.1f-1ubuntu2.27_all.deb  ...
正在解包 libssl-doc (1.0.1f-1ubuntu2.27) ...
正在处理用于 man-db (2.6.7.1-1ubuntu1) 的触发器 ...
正在设置 zlib1g-dev:i386 (1:1.2.8.dfsg-1ubuntu1.1) ...
正在设置 libssl-dev:i386 (1.0.1f-1ubuntu2.27) ...
正在设置 libssl-doc (1.0.1f-1ubuntu2.27) ...
liaozz@liaozz-VirtualBox:~$
```

1.2 使用 OpenSSL 的常用命令

A. 运行 openssl help，得到标准命令，数字摘要命令和加密命令

```
Standard commands
asn1parse         ca                ciphers           cms
crl               crl2pkcs7         dgst              dh
dhparam           dsa               dsaparam          ec
ecparam           enc               engine            errstr
gendh             gendsa            genpkey           genrsa
nseq              ocsp              passwd            pkcs12
pkcs7             pkcs8             pkey              pkeyparam
pkeyutl           prime             rand              req
rsa               rsautl            s_client          s_server
s_time            sess_id           smime             speed
spkac             srp               ts                verify
version           x509

Message Digest commands (see the `dgst' command for more details)
md4               md5               rmd160            sha
sha1

Cipher commands (see the `enc' command for more details)
aes-128-cbc       aes-128-ecb       aes-192-cbc       aes-192-ecb
aes-256-cbc       aes-256-ecb       base64            bf
bf-cbc            bf-cfb            bf-ecb            bf-ofb
camellia-128-cbc  camellia-128-ecb  camellia-192-cbc  camellia-192-ecb
camellia-256-cbc  camellia-256-ecb  cast              cast-cbc
cast5-cbc         cast5-cfb         cast5-ecb         cast5-ofb
des               des-cbc           des-cfb           des-ecb
des-ede           des-ede-cbc       des-ede-cfb       des-ede-ofb
des-ede3          des-ede3-cbc      des-ede3-cfb      des-ede3-ofb
des-ofb           des3              desx              rc2
rc2-40-cbc        rc2-64-cbc        rc2-cbc           rc2-cfb
rc2-ecb           rc2-ofb           rc4               rc4-40
seed              seed-cbc          seed-cfb          seed-ecb
seed-ofb
```

B. 如果对某个命令的用法不是很清楚，可以用“openssl 命令名称-help”查看该命令的说明。如输入“openssl passwd -help”，结果如下

```
liaozz@liaozz-VirtualBox:~$ openssl passwd -help
Usage: passwd [options] [passwords]
where options are
-crypt            standard Unix password algorithm (default)
-1                MD5-based password algorithm
-apr1             MD5-based password algorithm, Apache variant
-salt string      use provided salt
-in file          read passwords from file
-stdin            read passwords from stdin
-noverify         never verify when reading password from terminal
-quiet            no warnings
-table            format output as table
-reverse          switch table columns
```

C. 显示版本和编译参数:openssl version -a

```
liaozz@liaozz-VirtualBox:~$ openssl version -a
OpenSSL 1.0.1f 6 Jan 2014
built on: Tue Dec  4 20:10:05 UTC 2018
platform: debian-i386
options:  bn(64,32) rc4(8x,mmx) des(ptr,risc1,16,long) blowfish(idx)
compiler: cc -fPIC -DOPENSSL_PIC -DOPENSSL_THREADS -D_REENTRANT -DDSO_DLFCN -DHAVE_DLFCN_H -DL
-security -D_FORTIFY_SOURCE=2 -Wl,-Bsymbolic-functions -Wl,-z,relro -Wa,--noexecstack -Wall -D
DSHA1_ASM -DSHA256_ASM -DSHA512_ASM -DMD5_ASM -DRMD160_ASM -DAES_ASM -DVPAES_ASM -DWHIRLPOOL_A
OPENSSLDIR: "/usr/lib/ssl"
```

D. 查看 SSL 密码组合列表:openssl ciphers

```
liaozz@liaozz-VirtualBox:~$ openssl ciphers
ECDHE-RSA-AES256-GCM-SHA384:ECDHE-ECDSA-AES256-GCM-SHA384:ECDHE-RSA-AES256-SHA384:ECDHE-ECDSA-
AES256-SHA384:ECDHE-RSA-AES256-SHA:ECDHE-ECDSA-AES256-SHA:SRP-DSS-AES-256-CBC-SHA:SRP-RSA-AES-
256-CBC-SHA:SRP-AES-256-CBC-SHA:DHE-DSS-AES256-GCM-SHA384:DHE-RSA-AES256-GCM-SHA384:DHE-RSA-AE
S256-SHA256:DHE-DSS-AES256-SHA256:DHE-RSA-AES256-SHA:DHE-DSS-AES256-SHA:DHE-RSA-CAMELLIA256-SH
A:DHE-DSS-CAMELLIA256-SHA:ECDH-RSA-AES256-GCM-SHA384:ECDH-ECDSA-AES256-GCM-SHA384:ECDH-RSA-AES
256-SHA384:ECDH-ECDSA-AES256-SHA384:ECDH-RSA-AES256-SHA:ECDH-ECDSA-AES256-SHA:AES256-GCM-SHA38
4:AES256-SHA256:AES256-SHA:CAMELLIA256-SHA:PSK-AES256-CBC-SHA:ECDHE-RSA-DES-CBC3-SHA:ECDHE-ECD
SA-DES-CBC3-SHA:SRP-DSS-3DES-EDE-CBC-SHA:SRP-RSA-3DES-EDE-CBC-SHA:SRP-3DES-EDE-CBC-SHA:EDH-RSA
-DES-CBC3-SHA:EDH-DSS-DES-CBC3-SHA:ECDH-RSA-DES-CBC3-SHA:ECDH-ECDSA-DES-CBC3-SHA:DES-CBC3-SHA:
PSK-3DES-EDE-CBC-SHA:ECDHE-RSA-AES128-GCM-SHA256:ECDHE-ECDSA-AES128-GCM-SHA256:ECDHE-RSA-AES12
8-SHA256:ECDHE-ECDSA-AES128-SHA256:ECDHE-RSA-AES128-SHA:ECDHE-ECDSA-AES128-SHA:SRP-DSS-AES-128
-CBC-SHA:SRP-RSA-AES-128-CBC-SHA:SRP-AES-128-CBC-SHA:DHE-DSS-AES128-GCM-SHA256:DHE-RSA-AES128-
GCM-SHA256:DHE-RSA-AES128-SHA256:DHE-DSS-AES128-SHA256:DHE-RSA-AES128-SHA:DHE-DSS-AES128-SHA:D
HE-RSA-SEED-SHA:DHE-DSS-SEED-SHA:DHE-RSA-CAMELLIA128-SHA:DHE-DSS-CAMELLIA128-SHA:ECDH-RSA-AES1
28-GCM-SHA256:ECDH-ECDSA-AES128-GCM-SHA256:ECDH-RSA-AES128-SHA256:ECDH-ECDSA-AES128-SHA256:ECD
H-RSA-AES128-SHA:ECDH-ECDSA-AES128-SHA:AES128-GCM-SHA256:AES128-SHA256:AES128-SHA:SEED-SHA:CAM
ELLIA128-SHA:PSK-AES128-CBC-SHA:ECDHE-RSA-RC4-SHA:ECDHE-ECDSA-RC4-SHA:ECDH-RSA-RC4-SHA:ECDH-EC
DSA-RC4-SHA:RC4-SHA:RC4-MD5:PSK-RC4-SHA:EDH-RSA-DES-CBC-SHA:EDH-DSS-DES-CBC-SHA:DES-CBC-SHA
```

E. 测试所有算法速度：openssl speed（截取部分图）

```
liaozz@liaozz-VirtualBox:~$ openssl speed
Doing md4 for 3s on 16 size blocks: 3330648 md4's in 2.41s
Doing md4 for 3s on 64 size blocks: 4201911 md4's in 2.31s
Doing md4 for 3s on 256 size blocks: 2282845 md4's in 2.32s
Doing md4 for 3s on 1024 size blocks: 2041285 md4's in 2.73s
Doing md4 for 3s on 8192 size blocks: 252980 md4's in 2.42s
Doing md5 for 3s on 16 size blocks: 3593575 md5's in 2.42s
Doing md5 for 3s on 64 size blocks: 1978783 md5's in 2.25s
Doing md5 for 3s on 256 size blocks: 2183791 md5's in 2.60s
Doing md5 for 3s on 1024 size blocks: 1280375 md5's in 2.58s
Doing md5 for 3s on 8192 size blocks: 84328 md5's in 2.26s
Doing hmac(md5) for 3s on 16 size blocks: 2813761 hmac(md5)'s in 2.38s
Doing hmac(md5) for 3s on 64 size blocks: 2608527 hmac(md5)'s in 2.51s
Doing hmac(md5) for 3s on 256 size blocks: 1232062 hmac(md5)'s in 2.41s
Doing hmac(md5) for 3s on 1024 size blocks: 868380 hmac(md5)'s in 2.44s
Doing hmac(md5) for 3s on 8192 size blocks: 184423 hmac(md5)'s in 2.49s
Doing sha1 for 3s on 16 size blocks: 2971339 sha1's in 2.50s
Doing sha1 for 3s on 64 size blocks: 3566246 sha1's in 2.41s
Doing sha1 for 3s on 256 size blocks: 2450205 sha1's in 2.38s
Doing sha1 for 3s on 1024 size blocks: 622924 sha1's in 2.26s
```

F. 产生 RSA 密钥对：openssl genrsa -out 1.key 1024

```
liaozz@liaozz-VirtualBox:~$ openssl genrsa -out 1.key 1024
Generating RSA private key, 1024 bit long modulus
.++++++
.....................................++++++
e is 65537 (0x10001)
```

G. 取出 RSA 公钥:openssl rsa -in 1.key -pubout -out 1.pubkey

```
liaozz@liaozz-VirtualBox:~$ openssl rsa -in 1.key -pubout -out 1.pubout
writing RSA key
```

H. 加密文件:openssl enc -e -rc4 -in 1.key -out 1.key.enc

```
liaozz@liaozz-VirtualBox:~$ openssl enc -e -rc4 -in 1.key -out 1.key.enc
enter rc4 encryption password:
Verifying - enter rc4 encryption password:
liaozz@liaozz-VirtualBox:~$ ls
1.key      1.pubout          公共的  视频  文档  音乐
1.key.enc  examples.desktop  模板    图片  下载  桌面
```

I. 解密文件:openssl enc -d -rc4 -in 1.key.enc -out 1.key.dec

```
liaozz@liaozz-VirtualBox:~$ openssl enc -d -rc4 -in 1.key.enc -out 1.key.dec
enter rc4 decryption password:
liaozz@liaozz-VirtualBox:~$ ls
1.key      1.key.enc  examples.desktop  模板  图片  下载  桌面
1.key.dec  1.pubout   公共的            视频  文档  音乐
```

J. 计算文件的 MD5 值:openssl md5 1.key

```
liaozz@liaozz-VirtualBox:~$ openssl md5 1.key
MD5(1.key)= f9251df14a053cd2d756bf81c202f7a3
```

K. 计算文件的 SHA1 值:openssl sha1 1.key

```
liaozz@liaozz-VirtualBox:~$ openssl sha1 1.key
SHA1(1.key)= 5d4bad6148e6db2ed0519fcf7430ee1a834694e4
```

L. 计算 cryptoDemo.cpp 的 MD5 值，结果如下

```
root@liaozz-VirtualBox:/mnt/windows_share/lab2/cryptoDemo# openssl md5 cryptoDem
o.cpp
MD5(cryptoDemo.cpp)= ad42f4cc1cf8dbc31d6c8dbe46c784b8
```

2. 利用 OpenSSL 编程实现 AES 的加密、解密

- 测试用例（在 Linux 环境下，需要将第 10 行改为#include "/usr/include/openssl/aes.h"），测试成功

```
root@liaozz-VirtualBox:/mnt/windows_share/lab2/cryptoDemo# gcc -o cryptoDemo cryptoDemo.cpp -lcrypto
root@liaozz-VirtualBox:/mnt/windows_share/lab2/cryptoDemo# ./cryptoDemo
test success
The original string is:
  This is a sample. I am a programer.
 The encrypted string is:
  ◆"/◆◆◆$&◆◆◆ 96C◆H
 The decrypted string is:
  This is a sample. I am a programer.
```

- 修改例程 cryptoDemo.cpp 为 encfile.cpp：从命令行接受 3 个字

符串类型的参数：参数 1，参数 2，参数 3。参数 1=enc 表示加密，参数 1=dec 表示解密；参数 2 为待加密、解密的文件名；参数 3 为密码。

A. 增加 main 函数参数，根据参数进行解密与加密

```
int main(int argc, char* argv[])
{
    if(strcmp(argv[1],"enc")==0)
        EncFile(argv[2],argv[3],strlen(argv[3]));
    else if(strcmp(argv[1],"dec")==0)
        DecFile(argv[2],argv[3],strlen(argv[3]));
    else{
        printf("ERROR!\n");
    }
    return 0;
```

B. 对于加密，每次从文件中读取 16 字节的数据到 buf 中，然后将加密后的数据写入到加密文件，对于文件末尾不足 16 字节的部分，也需将数据补全到 16 位进行加密，然后写入到加密文件中。为了后续解密能完全恢复文件，在加密文件末尾附上一个数，其表示原文件最后一个数据块实际有多少字节。

```
    while (!feof(fp_in)){
        memset(buf,0,16);
        ReadLen=fread(buf, sizeof(unsigned char),16,fp_in);
//      fwrite(buf,sizeof(unsigned char ),ReadLen,fp_test);
//      if(ReadLen<16)
//              printf("ReadLen=%d\n",ReadLen);
        AES_encrypt(buf,buf2,&aeskey);
        fwrite(buf2, sizeof(unsigned char),16,fp_out);
    }
        buf2[0]=(unsigned char)ReadLen;
//      printf("@@@%d@@@\n",(int)buf2[0]);
        fwrite(buf2,sizeof(unsigned char),1,fp_out);
```

C. 对于解密，首先从加密文件末尾读出最后一个字节的数据，其表示加密文件最后一个 16 字节数据块实际有多少字节是有效的，然后对加密文件的 16 字节块依次解密，对于最后一部分，只将有效的数据写入解密文件。

```
    memset(buf,0,16);
    fseek(fp_in,-1L,2);
    fread(buf,sizeof(unsigned char),1,fp_in);
    EndLen=(int)buf[0];
    fseek(fp_in,0L,0);
     memset(buf,0,16);
     ReadLen=fread(buf, sizeof(unsigned char),16,fp_in);
     for(i=0;i<16;i++)
         buf3[i]=buf[i];
     while((ReadLen=fread(buf, sizeof(unsigned char),16,fp_in))!=1){
         AES_decrypt(buf3,buf2,&aeskey);
         fwrite(buf2,sizeof(unsigned char),16,fp_out);
         for(i=0;i<16;i++)
                 buf3[i]=buf[i];
         memset(buf,0,16);
}
         AES_decrypt(buf3,buf2,&aeskey);
         fwrite(buf2,sizeof(unsigned char),EndLen,fp_out);
```

D. 使用密钥 123abc 对文件进行加密解密

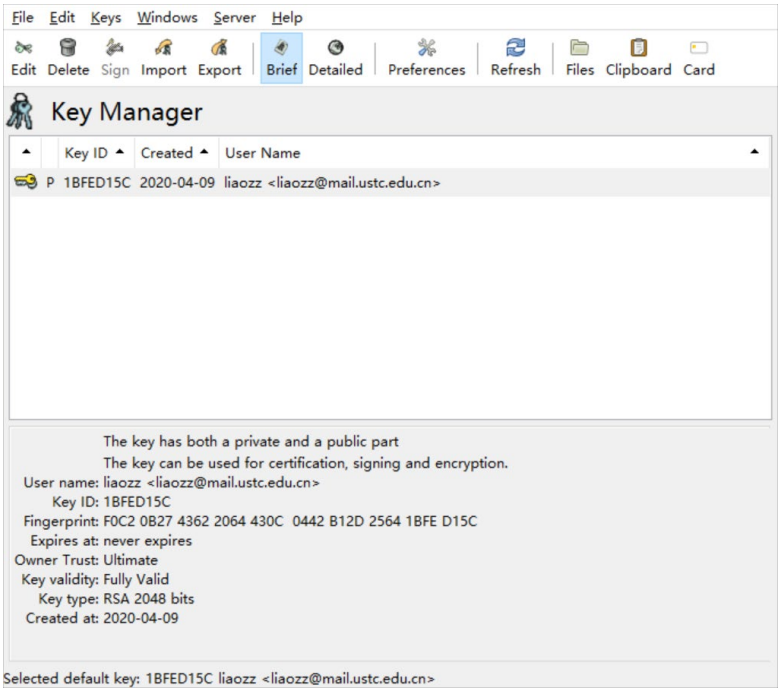
```
liaozz@liaozz-VirtualBox:~/lab$ ./encfile enc InfoSecIntro.htm 123abc
liaozz@liaozz-VirtualBox:~/lab$ ./encfile dec InfoSecIntro.htm.enc 123abc
```

E. 检查解密后的文件与原文件是否相同，计算原文件与解密文件的 MD5 值，发现值相等，说明算法正确

```
liaozz@liaozz-VirtualBox:~/lab$ openssl md5 InfoSecIntro.htm
MD5(InfoSecIntro.htm)= f812d23cb3cc0bb554dfc3c0c7794a00
liaozz@liaozz-VirtualBox:~/lab$ openssl md5 dec-InfoSecIntro.htm
MD5(dec-InfoSecIntro.htm)= f812d23cb3cc0bb554dfc3c0c7794a00
```

3. 用 PGP 实现加密和解密

- 步骤 1 ：产生一对 RSA 密钥，以 passphrase=“2020ustc44”生成密钥



- 步骤 2：互换公钥，将公钥导出(Export Keys)到一个文件中(假定文件名为 pub_key.key)，传递给需要给自己发送加密文件的电脑。
- 步骤 3：向对方发送加密文件，后续实验需要两人，不再进行。

四、 实验总结

1. 在 Linux 环境下配置了 OpenSSL
2. 熟悉使用了 OpenSSL 的命令
3. 练习了在 C 程序中使用 OpenSSL 的方法
4. 练习了 PGP 的使用