实验项目：密码应用-文件安全传输

【实验目的】

1.掌握安全通信中常用的加密算法

2.掌握数字签名过程

3.掌握文件安全传输的基本步骤

【实验内容】

A 为发送方，B 为接收方。自行设计实验步骤，采用对称加密算法、非对称加密算法和哈希算法相结合的方式，通过使用密码工具实现文件信息的安全传输。

【实验原理】

1. 数字签名

传统的签名在商业和生活中广泛使用，它主要作为身份的证明手段。在现代的网络活动中，人们希望把签名制度引入到网络商业和网络通信的领域，用以实现身份的证明。PKI（公钥基础设施）体系利用数字签名技术来保证信息传输过程中的数据完整性以及提供对信息发送者身份的认证和不可抵赖性。

数字签名的过程如图 4.1 所示。

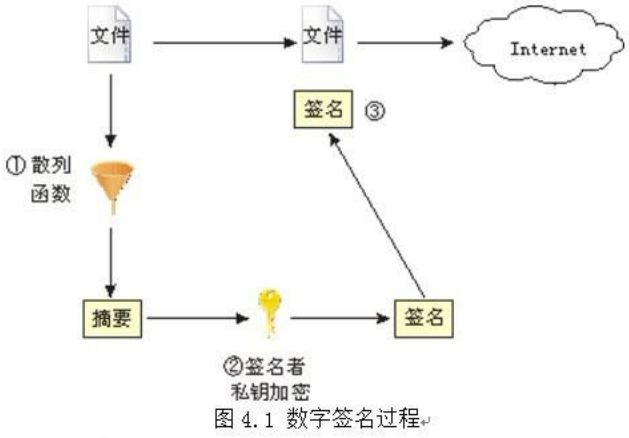

图 4.1 数字签名过程

2．安全通信要求

保密要求：终端 A 的信息（明文）要加密通信给终端 B。

认证要求：终端 B 能认证发送人的身份是终端 A，而不是其他人。

数据完整性要求：终端 B 能认证收到的密文没有被篡改。

不可否认要求：终端 A 事后不能否认曾经把信息传递给终端 B。

以上这些要求也是真实世界中安全通信的基本要求。

【实验步骤】

1. 发送方文件安全传输和接收方安全接收设计

实验应采用对称加密算法、非对称加密算法和哈希算法相结合的方式，通过使用密码工具实现信息的安全传输。以终端 A 为发送方，终端 B 为接收方为

例，实现流程大致应如下。

A 操作：

（1）与 B 预先协商好通信过程中所使用到的对称加密算法、非对称加密算法和哈希函数；

该实验所使用的分别为：AES 加密算法，RSA 非对称加密算法，MD5 哈希函数。

（2）采用对称加密算法（密钥称之为会话密钥）对传输信息进行加密得到密文，确保传输信息的保密性；

```
void encrypt_aes(string& mes)
{
    encrypt_ecb(S_BOX,mes);
}
void decrypt_aes(string& mes){
    decrypt_ecb(S_BOX, N_S_BOX,mes);
}
```

```
cout << "------用户A加密明文------" << endl;
//生成密文文件test.txt
encrypt_aes(mes);
```

（3）使用 B 的公钥对会话密钥进行加密，确保传输信息的保密性以及信息接收方的不可否认性；

```
void init(){
    //公钥系统建立 AB产生自己的公私钥对
    cout << "------建立Alice公私钥对------" << endl;
    int p1,q1;
    Make_PublicKey_And_PrivateKey(p1,q1,n1,e1,d1);
    cout << "------建立Bob公私钥对------" << endl;
    int p2,q2;
    Make_PublicKey_And_PrivateKey(p2,q2,n2,e2,d2);
    cout << "------公钥系统已建立------" << endl;
}
```

```
cout << endl <<"------用户A加密对称密钥key------" << endl;
//加密后的会话密钥  key.txt
Encrypt_key(key,e2,n2,C1);
saveFile_int(C1);
```

（4）采用哈希函数（生成文件摘要）确保传输信息的完整性，并使用自己的私钥对文件摘要进行签名（得到数字签名），确保信息发送方的不可否认性；

```
//哈希+数字签名  promise.txt
cout << "------用户A对明文做哈希处理------" << endl;
digest = md5(mes);
Encrypt_key(digest,d1,n1,C2);
saveFile_int(C2);
```

（5）将密文、加密后的会话密钥和数字签名打包封装（放到一起）后，通过网络传输给 B。

B 操作：

（1）与 A 预先已协商好通信过程中所使用到的对称加密算法、非对称加密算法和哈希函数；

（2）使用自己的私钥对 A 加密的会话密钥进行解密，得到准会话密钥；

```
// 私钥对会话密钥进行解密 key.txt
cout << "------用户B对会话密钥进行解密------" << endl;
readFile_int(C1);
Decrypt_key(C1,d2,n2,key);
cout << "解得的密钥为:" << endl;
cout << key << endl;
```

```
void Decrypt_key(vector<int> C, int d, int n, string& show)
{
    for(int t : C){
        int product = 1;//product表示c自身不断相乘的积
        for (int i = 1; i <= d; ++i)//用d来控制c的指数
            {product = (product * t) %n;}//注意不要忘记%n
        show.push_back(product+'a'-1);
    }
}
```

（3）使用准会话密钥对得到的密文进行解密，得到准明文；

```
//对得到的密文进行解密  test.txt
cout << "------用户B对得到的密文进行解密------" << endl;
decrypt_aes(mes);
```

（4）使用 A 的公钥对得到的数字签名进行签名验证，得到准明文摘要；

```
//数字签名解密 promise.txt
cout << "------用户B对数字签名解密------" << endl;
readFile_int(C2);
Decrypt_key(C2,e1,n1,digest);
```

（5）使用哈希函数计算得到准明文摘要；

```
// 哈希运算
digest1 = md5(mes);
if(digest == digest1)
    cout << "文件安全传输成功! " << endl;
else
    cout << "文件传输失败! " << endl;
return true;
```

（6）将计算得到的摘要与准明文摘要进行比较，若相同则表明文件安全传输成功。

【实验结果】

```
PS D:\Test\cworkspace\cryptography\8>  & 'c:\Users\Hui
ebugAdapters\bin\WindowsDebugLauncher.exe' '--stdin=Mi
ngine-Out-b3ldlfdo.0mf' '--stderr=Microsoft-MIEngine-E
.qeb' '--dbgExe=D:\x86_64-8.1.0-release-win32-seh-rt_v
------用户A加密明文------

请输入要加密的文件名，该文件必须和本程序在同一个目录
ming.txt

请输入32位长度密钥:
akhvsrmfsyzibetagmhmademjkliddpq

加密后的密文为:
c1 76 4c c 76 be 5f eb 72 d7 ac f7 da 42 7c de
请将密文写进文件，比如'test.txt':
test.txt
已经将密文写进test.txt中了,可以在当前目录中找到它。

------用户A加密对称密钥key------
信息已加密(RSA)!
请将密文写进文件，比如'key.txt'/'promise.txt':
key.txt
已经将密文写进key.txt中了,可以在当前目录中找到它
------用户A对明文做哈希处理------
哈希值: f25a2fc72690b780b2a14e140ef6a9e0
信息已加密(RSA)!
请将密文写进文件，比如'key.txt'/'promise.txt':
pro.txt
akhvsrmfsyzibetagmhmademjkliddpq
```

```
------用户B对得到的密文进行解密------
请输入要解密的文件名，该文件必须和本程序在同一个目录
test.txt

请输入32位长度密钥:
akhvsrmfsyzibetagmhmademjkliddpq

解密结果为:
iloveyou
------用户B对数字签名解密------
请输入文件名:
pro.txt

文件内容为:
73 -492 -450 1 -492 73 434 -105 -492 -652 -381 -20 657 -105 -398 -20 657 -492 1 -346 -198 125 -346 -198 -20 125
 73 -652 1 -381 125 -20
哈希值: f25a2fc72690b780b2a14e140ef6a9e0
文件安全传输成功!
B已经完成解密!
```

实验中，Alice 与 Bob 约定 AES\RSA\MD5 加密方法。Alice 用会话密钥将明文 ming.txt 进行加密，存入 test.txt 文件中；再用 Bob 的公钥对会话密钥进行加密存入 key.txt；最后将明文进行 MD5 操作得到摘要，用自己的私钥加密存入 pro.txt 文件中。

Bob 利用自己的私钥对 key.txt 文件解密得到会话密钥；用会话密钥对 test.txt 文件解密得到明文。将 pro.txt 文件用 Alice 公钥解密得到摘要，将明文进行 MD5 操作并进行比较，当两结果相同时，文件安全传输，实验完成。