流程如下图：



主机A： 发送方

主机B： 接收方

**发送部分：**

A的DES公钥为“qwertyui“， A利用该公钥将“财务报表.doc”加密为“财务报表\_加密.doc“

A从B处获得B的RSA公钥，对DES秘钥（qwertyui）使用RSA公钥加密，结果（256字节）存放在“财务报表\_加密.doc“的文件尾部。

A使用自己的RSA私钥对文件进行签名，生成256位的签名数据， 存放在“财务报表\_加密.doc“的文件尾部。

A将“财务报表\_加密.doc“文件发送给B



**接收部分：**

B从A处“财务报表\_加密.doc“（文件长度为doc\_len），获取文件doc\_len-512至doc\_len-256位置的数据，使用B的RSA私钥解密，获得DES秘钥。

使用DES私钥解密文件0至doc\_len-512字节数据，得到原始文件，保存为“财务报表.doc“

从A处获取A的RSA公钥，对“财务报表\_加密.doc “前doc\_len-256字节的数据处理并与文件doc\_len-256至doc\_len处的签名验证，若验证成功则显示”通过签名验证“， 否则显示”未通过签名验证“



安全性：

使用对称加密（DES）与非对称加密（RSA）

数据采用DES加密，保证数据的安全。

DES加密的密钥使用RSA传输，避免密钥在传输过程中泄露

数字签名的不扣否认性：

签名发布者生成私钥和公钥，使用私钥生成签名，公钥可验证签名。

因此签名发布者对需要发布的文件使用私钥签名，然后将公钥发布出去，因此其他人可以使用公钥验证是否该文件是否为发布者发布。

此外，签名过程包括了“报文摘要“的过程，该过程使用MD5实现，是不可逆的。因此文件稍作改动，MD5的结果都会不一样，保证数据的完整性。

运行结果：

进入文件夹A

编译程序A



运行A



进入B文件夹

编译B



运行B

