密码学基础实验

一、实验原理

1.1 DES 加解密

DES 全称为 Data Encryption Standard，即数据加密标准，是一种使用密钥加密的块算法，1977 年被美国联邦政府的国家标准局确定为联邦资料处理标准（FIPS），并授权在非密级政府通信中使用，随后该算法在国际上广泛流传开来。

对称性：DES 是对称的，也就是说它使用同一个密钥来加密和解密数据。与此相对的是 RSA 加密算法，是一种非对称加密算法

分组性：DES 还是一种分组加密算法，该算法每次处理固定长度的数据段，称之为分组。DES 分组的大小是 64 位，如果加密的数据长度不是 64 位的倍数，可以按照某种具体的规则来填充位。

“混乱和扩散”的原则：混乱的目的是为隐藏任何明文同密文、或者密钥之间的关系，而扩散的目的是使明文中的有效位和密钥一起组成尽可能多的密文。两者结合到一起就使得安全性变得相对较高。

DES 算法具体通过对明文进行一系列的排列和替换操作来将其加密：过程的关键就是从给定的初始密钥中得到 16 个子密钥的函数。要加密一组明文，每个子密钥按照顺序（1-16）以一系列的位操作施加于数据上，每个子密钥一次，一共重复 16 次。每一次迭代称之为一轮。要对密文进行解密可以采用同样的步骤，只是子密钥是按照逆向的顺序（16-1）对密文进行处理。

1.2 AES 加解密

AES 全称为 Advanced Encryption Standard，是美国联邦政府采用的一种区块加密标准，用来替代原先的 DES。

AES 加密过程涉及到 4 种操作：字节替代（Sub Bytes）、行移位（Shift Rows）、列混淆（Mix Columns）和轮密钥加（Add Round Key）。解密过程分别为对应的逆操作。由于每一步操作都是可逆的，按照相反的顺序进行解密即可恢复明文。

加解密中每轮的密钥分别由初始密钥扩展得到。算法中 16 字节的明文、密文和轮密钥都以一个 4×4 的矩阵表示。AES 的具体加密解密流程，如图 1.1 所示。

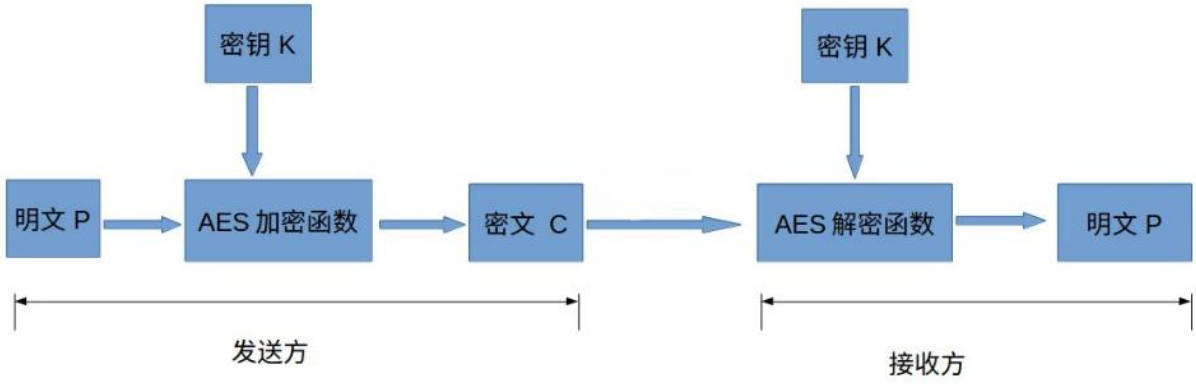


图 1.1 AES 加密解密流程

下面简单介绍下各个部分的作用与意义：

明文 P：

没有经过加密的数据。

密钥 K

用来加密明文的密码，在对称加密算法中，加密与解密的密钥是相同的。密钥为接收方与发送方协商产生，但不可以直接在网络上传输，否则会导致密钥泄漏，通常是通过非对称加密算法加密密钥，然后再通过网络传输给对方，或者直接面对面商量密钥。密钥是绝对不可以泄漏的，否则会被攻击者还原密文，窃取机密数据。

AES 加密函数

设 AES 加密函数为 E，则 C = E(K, P),其中 P 为明文，K 为密钥，C 为密文。也就是说，把明文 P 和密钥 K 作为加密函数的参数输入，则加密函数 E 会输出密文 C。

密文 C

经加密函数处理后的数据

AES 解密函数

设 AES 解密函数为 D，则 P = D(K, C),其中 C 为密文，K 为密钥，P 为明文。

也就是说，把密文 C 和密钥 K 作为解密函数的参数输入，则解密函数会输出明文 P。

1.3 RSA 加解密

RSA 加密算法，是世界上第一个非对称加密算法，也是数论的第一个实际应用。它的算法如下：

①找两个非常大的质数 p 和 q（通常 p 和 q 都有 155 十进制位或都有 512 十进制位）并计算 n=pq，k=(p-1)(q-1)。

②将明文编码成整数 M，保证 M 不小于 0 但是小于 n。

③任取一个整数 e，保证 e 和 k 互质，而且 e 不小于 0 但是小于 k。加密钥匙（称作公钥）是(e, n)。

④找到一个整数 d，使得 ed 除以 k 的余数是 1（只要 e 和 n 满足上面条件，d 肯定存在）。解密钥匙（称作密钥）是(d, n)。

加密过程：加密后的编码 C 等于 M 的 e 次方除以 n 所得的余数。

解密过程：解密后的编码 N 等于 C 的 d 次方除以 n 所得的余数。

只要 e、d 和 n 满足上面给定的条件。M 等于 N。

二、实验环境

2.1 硬件

Dell G3579 笔记本电脑。

2.2 软件

运行系统：Windows 10 Pro N for Workstations

开发工具：VSCode

编程语言：java

三、实验结果

3.1 DES 加解密实验

3.1.1 加解密字符串“abcdef”

代码：

```
import util.DES;
public class Main_DES{
    Run | Debug
    public static void main(String args[]){
        String es=DES.encrypt( password: "12345678", data: "abcdef");
        System.out.println("加密结果：");
        System.out.println(es);
        String ds=DES.decrypt( password: "12345678",es);
        System.out.println("解密结果：");
        System.out.println(ds);
    }
}
```

运行结果：

```
加密结果：
+Ty5ad/SjLw=
解密结果：
abcdef
```

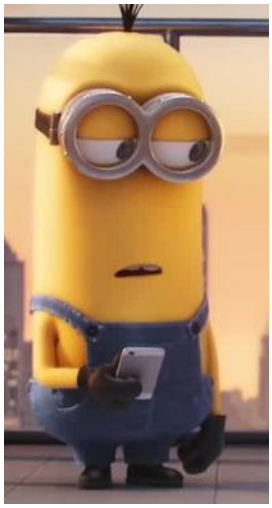
3.1.2 加解密图片

名称	修改日期	类型	大小
util	2022/4/11 19:50	文件夹	
1.png	2022/5/4 17:15	PNG 文件	79 KB
2.png	2022/5/4 17:14	PNG 文件	79 KB

代码：

```
import util.DES;
public class Main_DES{
    Run | Debug
    public static void main(String args[]){

        DES.encryptFile( password: "aaaaaaaa", srcFile: "1.png",
                         destFile: "2.png");
        DES.decryptFile( password: "aaaaaaaa", srcFile: "2.png", destFile: "1.png");
    }

}
```

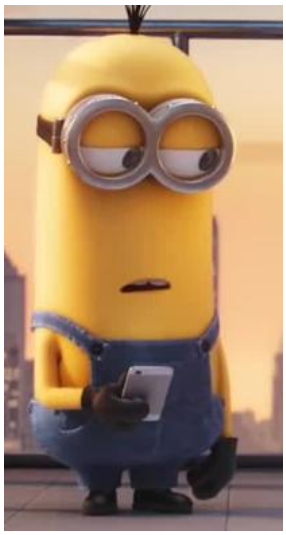
原始图片：



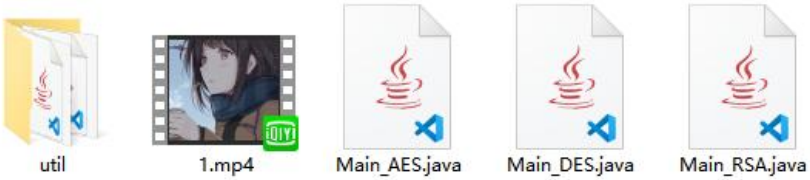
加密后：

2.png
似乎不支持此文件格式。

解密后：



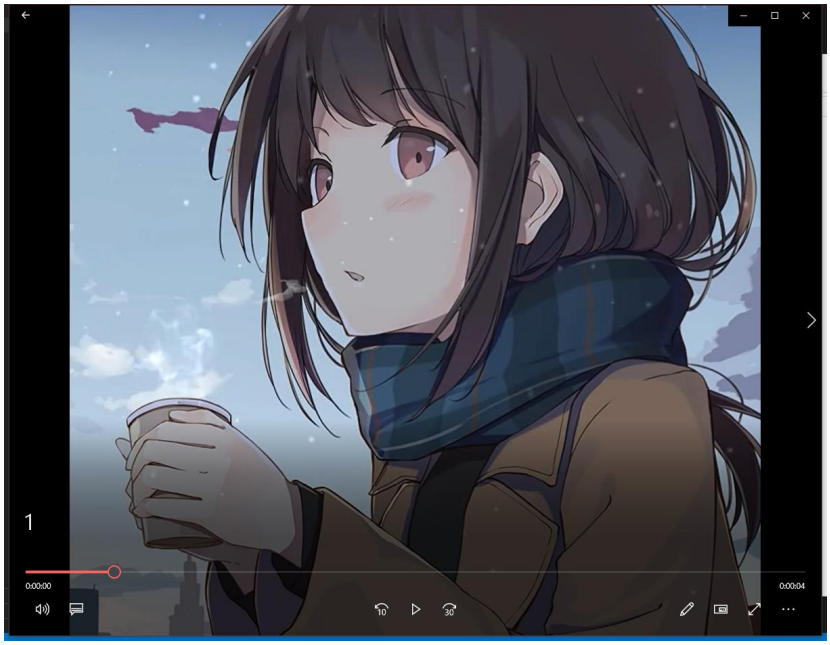
3.1.3 加解密视频



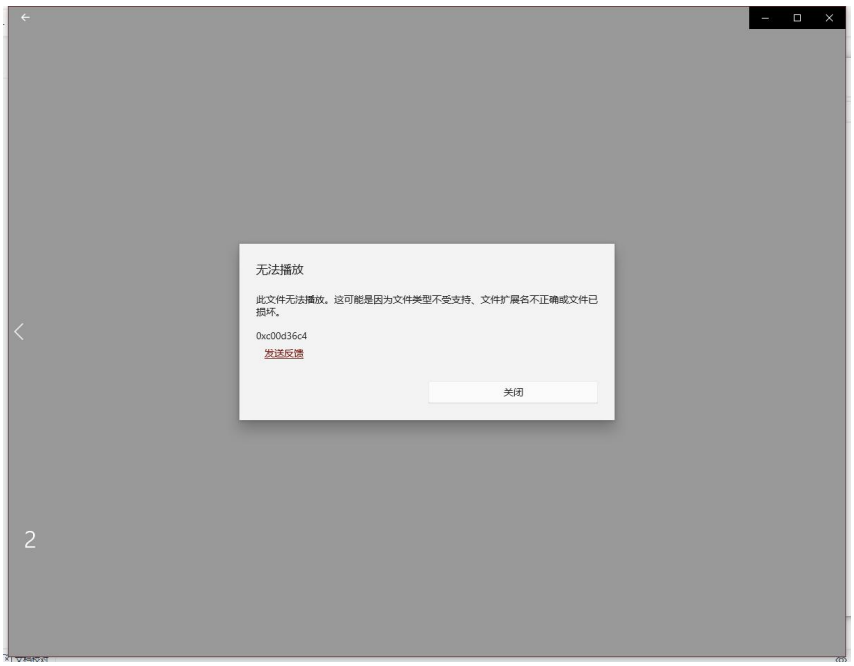
代码：

```
import util.DES;
public class Main_DES{
    Run | Debug
    public static void main(String args[]){

        DES.encryptFile( password: "aaaaaaaa", srcFile: "1.mp4",
                         destFile: "2.mp4");
        DES.decryptFile( password: "aaaaaaaa", srcFile: "2.mp4", destFile: "1.mp4");
    }

}
```

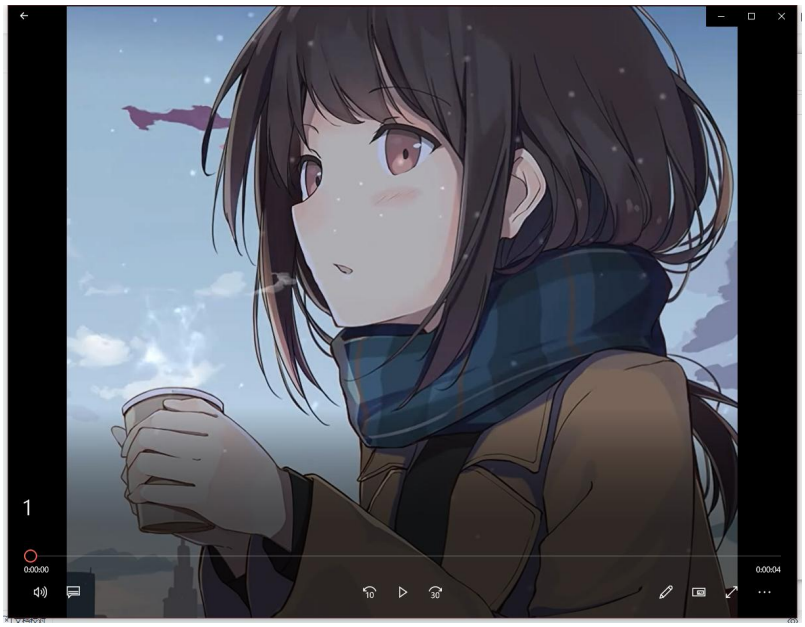
加密前：



加密后：



解密后：



3.2 AES 加解密实验

3.2.1 加解密字符串“helloworld”

代码

```java
public class Main_AES {
    Run | Debug
    public static void main(String args[]){
        String es=AES.encrypt( key: "aaaaaaaaaaaaaaaa", data: "helloworld");
        System.out.println("加密后：");
        System.out.println(es);
        String ds=AES.decrypt( key: "aaaaaaaaaaaaaaaa",es);
        System.out.println("解密后：");
        System.out.println(ds);

    }
}
```

运行结果：

```
加密后：
R76G1mkkKASECAiwKJFSYQ==
解密后：
helloworld
```

3.2.2 加解密图片

代码：

```java
public class Main_AES {
    Run | Debug
    public static void main(String args[]){

        AES.encryptFile( password: "aaaaaaaaaaaaaaaa", srcFile: "3.png", destFile: "4.png");
        AES.decryptFile( password: "aaaaaaaaaaaaaaaa", srcFile: "4.png", destFile: "3.png");
    }
}
```

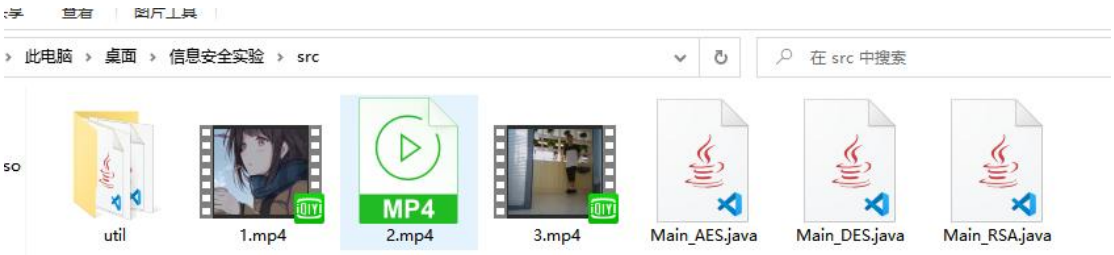
原始图片：

加密后：

4.png
似乎不支持此文件格式。

解密后：

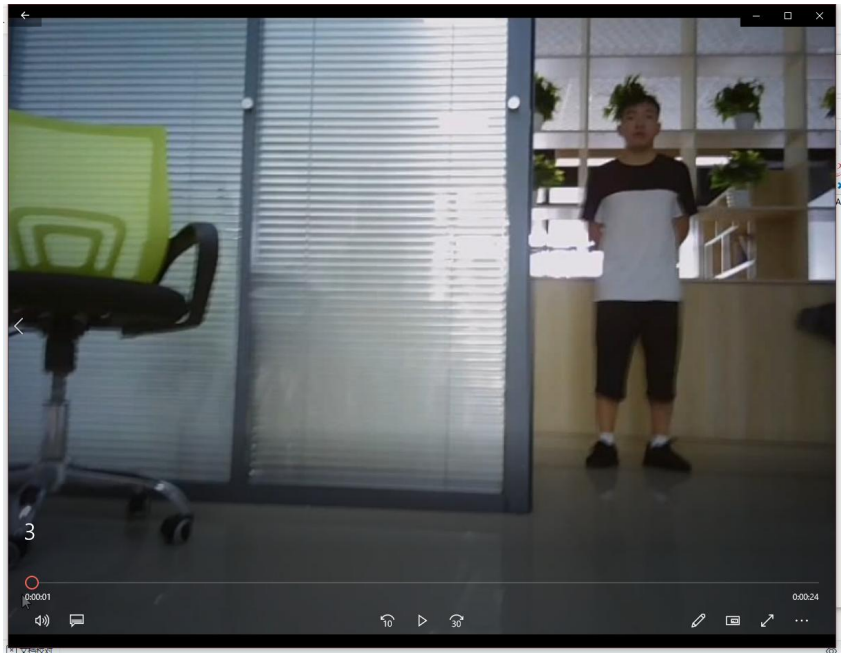
3.2.3 加解密视频



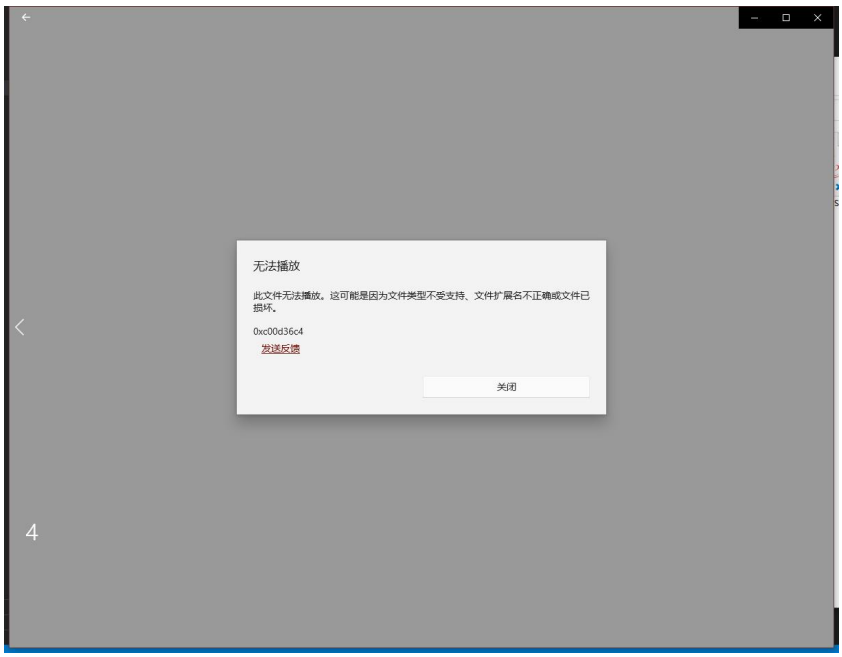
代码：

```
import util.AES;
public class Main_AES {
    Run | Debug
    public static void main(String args[]){

        AES.encryptFile( password: "aaaaaaaaaaaaaaaa",  srcFile: "3.mp4",  destFile: "4.mp4");
        AES.decryptFile( password: "aaaaaaaaaaaaaaaa",  srcFile: "4.mp4",  destFile: "3.mp4");
    }
}
```

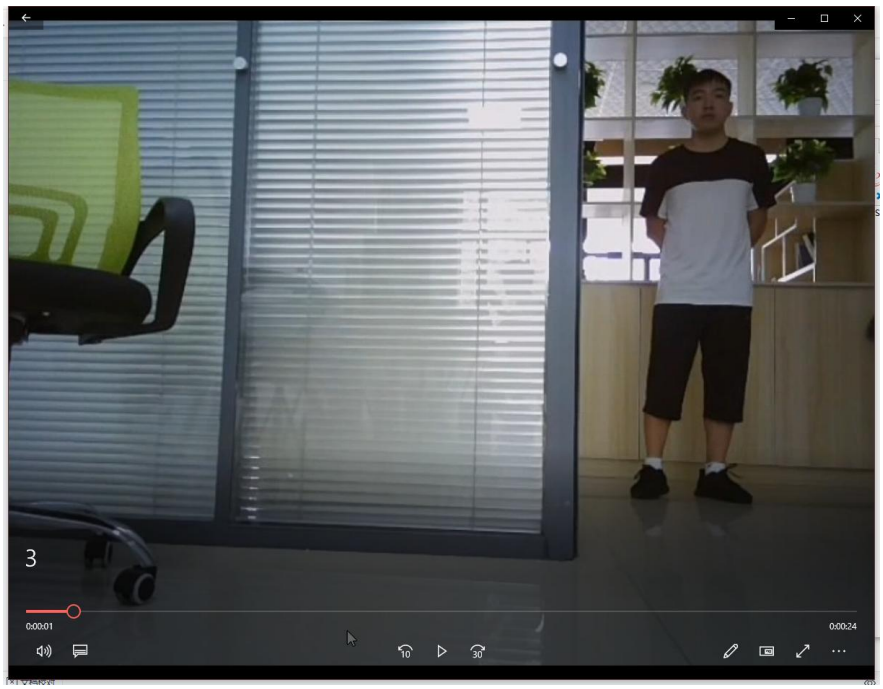
加密前：


3
0:00:01
0:00:24

加密后：


无法播放
此文件无法播放。这可能是因为文件类型不受支持、文件扩展名不正确或文件已损坏。
0xc00d36c4
发送反馈
关闭
4

解密后：


3
0:00:01
0:00:24

3.3 RSA 加解密实验

RSA 加密算法，是世界上第一个非对称加密算法，也是数论 的第一个实际应用。它的算法如下：

1.找两个非常大的质数 p 和 q（通常 p 和 q 都有 155 十进制位或都有 512 十进 制位）并计算 n=pq，k=(p-1)(q-1)。

2.将明文编码成整数 M，保证 M 不小于 0 但是小于 n。

3.任取一个整数 e，保证 e 和 k 互质，而且 e 不小于 0 但是小于 k。加密钥匙（称作公钥）是(e, n)。

4.找到一个整数 d，使得 ed 除以 k 的余数是 1（只要 e 和 n 满足上面条件，d 肯定存在）。解密钥匙（称作密钥）是(d, n)。

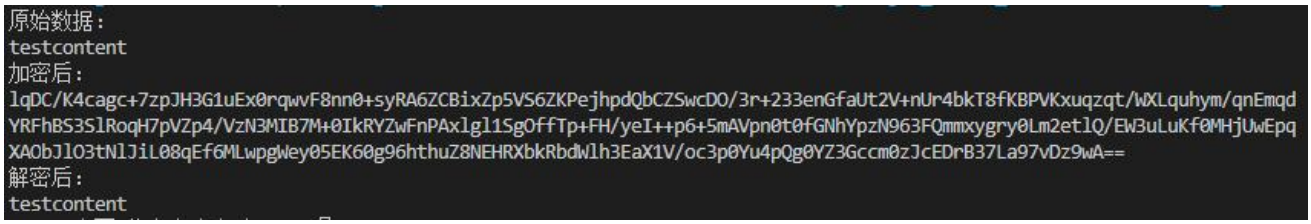
加密过程： 加密后的编码 C 等于 M 的 e 次方除以 n 所得的余数。

解密过程：解密后的编码 N 等于 C 的 d 次方除以 n 所得的余数。只要 e、d 和 n 满足上面给定的条件。M 等于 N。

3.3.1 代码

```
public class Main_RSA {
    Run | Debug
    public static void main(String args[]){
        try {
            Map<String, String> map= RSA.generateKeyPair();
            String pub=map.get("publicKey");
            String pri=map.get("privateKey");

            String data="testcontent";
            System.out.println("原始数据：");
            System.out.println(data);
            String en_result=RSA.encrypt(data,pub);
            System.out.println("加密后：");
            System.out.println(en_result);
            String de_result=RSA.decrypt(en_result, pri);
            System.out.println("解密后：");
            System.out.println(de_result);
            }
            catch (Exception e) {
                e.printStackTrace();
            }
    }
}
```

3.3.2 运行结果



非对称加密算法（如 RSA）同前文所述 DES、AES 等对称加密算法不同的是，对称加密算法的加密与解密密钥相同，而非对称加密算法的加密密钥与解密密钥不同。众所周知，RSA 加密算法基于一个十分简单的数论事实：将两个大素数相乘十分容易，但想要对其乘积进行因式分解却极其困难，因此可以将乘积公开作为加密密钥。