

# 文本加解密系统

陶蕊

东南大学 网络空间安全学院



2020年6月

# 目录

一、系统需求.....	3
二、系统框架.....	3
三、工作流程.....	4
四、系统功能模块.....	4
4.1 明文输入.....	4
4.1.1 字符串输入.....	5
4.1.2 文件输入.....	5
4.2 非对称加密.....	6
4.3 哈希签名.....	8
4.4 对称加密.....	8
4.5 发送密文.....	9
4.6 接收密文.....	9
4.7 非对称加密的解密.....	10
4.8 对称加密的解密.....	11
4.9 再次哈希签名.....	12
4.10 比较.....	13
五、界面设计.....	14
5.1 发送方 A 的界面.....	14
5.2 接收方 B 的界面.....	15
六、数据存储方式.....	15

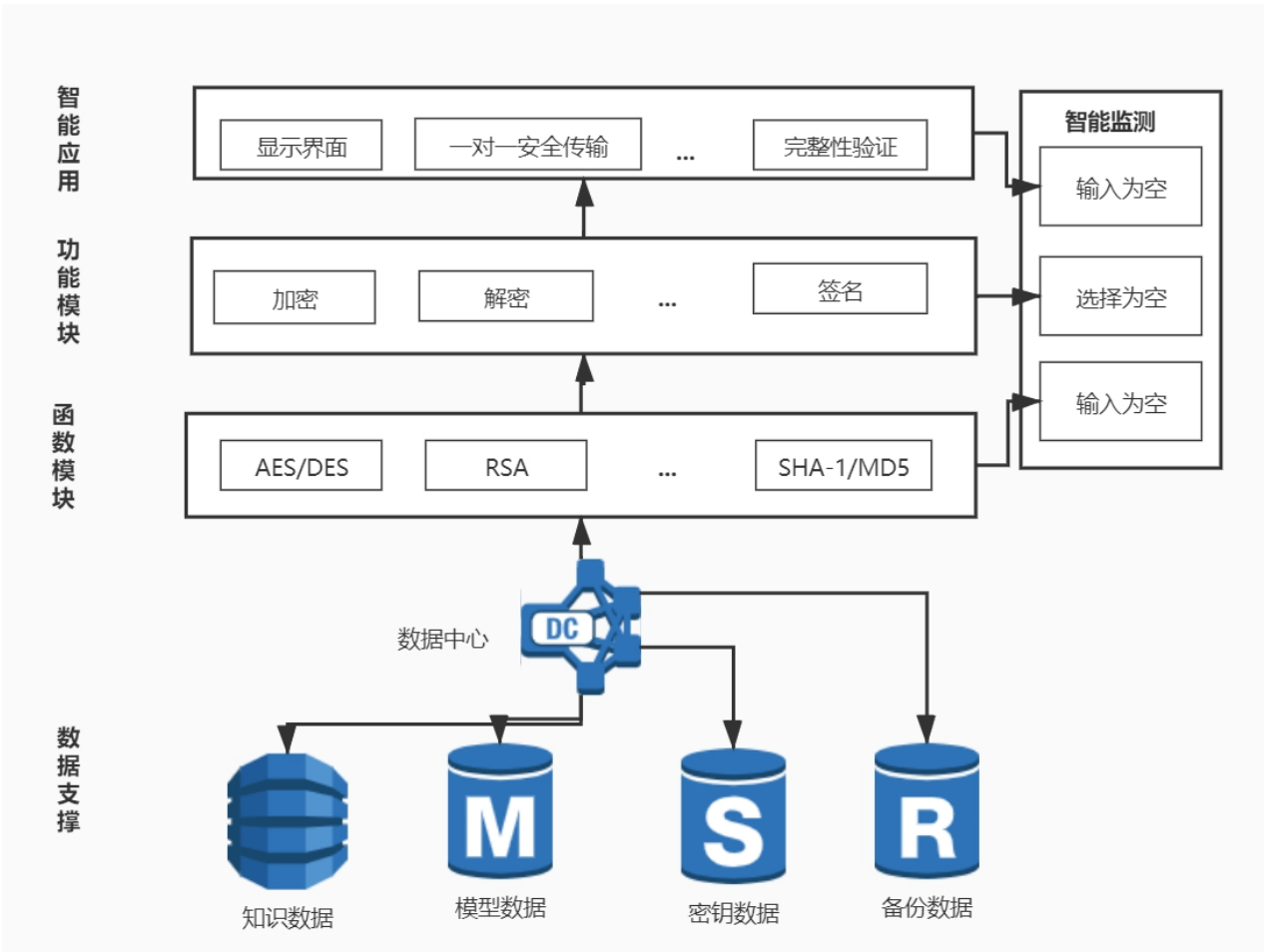
# 一、系统需求

该系统实现一个完整的从发送方A到发送方B的信息传输过程，其中包括对信息的签名、加密、组合、解密、验证签名等过程，实现信息的保密性和完整性，实现信息传输过程的安全需求。

对该系统的具体算法和内容需求如下：

- (1) 既可以对字符串进行签名和加密，也可对输入的文件内容进行签名和加密；
- (2) Hash 签名算法包含 SHA-1和MD5，在系统中可选；
- (3) 非对称加密算法为RSA算法，程序中能够产生不同的私钥和公钥对；密钥长度不得小于 200位；
- (4) 对称加密算法包含 DES和AES，在系统中可选；
- (5) 采用eclipse 4.8M5 平台开发。

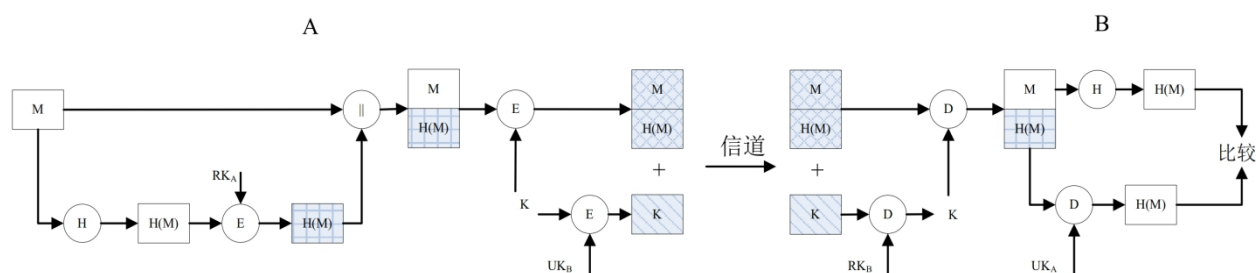
# 二、系统框架



图一 系统框架图

### 三、工作流程

下图二所示为系统的总体流程图：



图二 系统流程图

说明：

(1) M表示明文，H表示Hash 函数，E表示加密算法，D表示解密算法， $RK_A$  表示发送方A的私钥， $UK_A$ 表示发送方A的公钥， $RK_B$  表示发送方B的私钥， $UK_B$ 表示发送方B的公钥，||表示组合。

(2) 阴影部分表示加密后的结果。

发送方：首先将发送方A输入的明文进行哈希签名，用A的私钥对其进行非对称加密，称为信息一号；将明文与该信息一号进行组合，用对称加密算法将其加密为信息二号；将对称密钥K用B的公钥进行加密，称之为信息三号；将信息二号与信息三号组合为一个整体，通过安全信道发送给接收方B。

接收方：首先将信息二号与信息三号拆分开，将信息三号用B的私钥进行解密，求出对称密钥K；使用该密钥K对信息二号进行对称加密算法的解密，得到明文与信息一号的组合；将其组合拆开，得到明文，再对明文部分进行同样哈希签名，称之为信息四号；使用A的公钥对信息一号进行解密，求得发送方对明文的哈希签名，称为信息五号；将信息四号与信息五号进行比较，观察其是否相同。

### 四、系统功能模块

#### 4.1 明文输入

支持两种明文输入模式，分别为字符串直接输入模式和文件输入模式。

### 4.1.1 字符串直接输入模式



输入为字符串，内部函数该输入框的字符串。

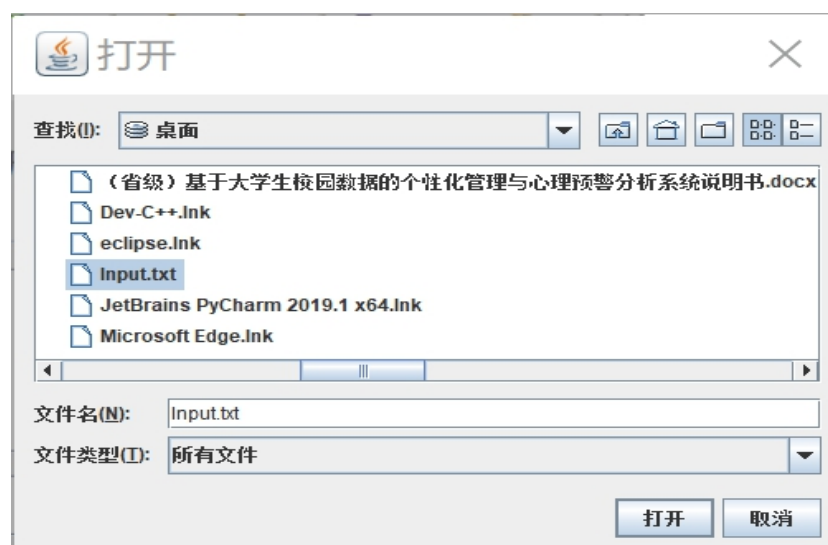
### 4.1.2 文件输入模式模式

点击选择文件，打开文本选择框，选择要输入的文件，需为.txt格式，输入框内显示文件的地址，系统后台可读取文件内容保存为字符串。

该接口函数为：

```
button_1.addActionListener(new ActionListener(){  
    public void actionPerformed(ActionEvent e) {}  
});
```

//点击按钮，弹出选择文件框，输入为选择的文件，输出为文件内的字符串内容。





## 4.2 非对称加密

非对称加密为RSA加密算法模块，可以生成素数 $p$ 、 $q$ ，模数 $n$ ，A、B的公钥 $e$ 和私钥 $d$ ，生成后点击保存，分别将模数 $n$ ，A、B的公钥 $e$ 和私钥 $d$ 保存为.txt格式。

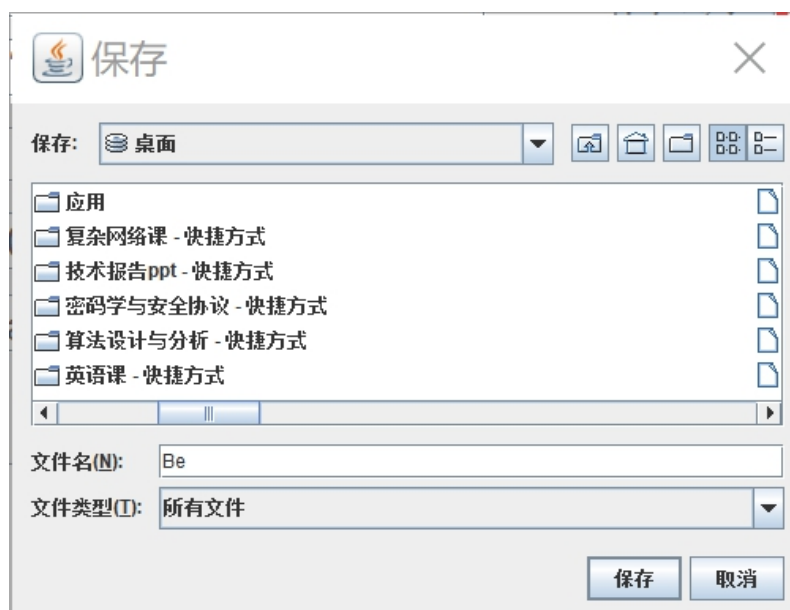
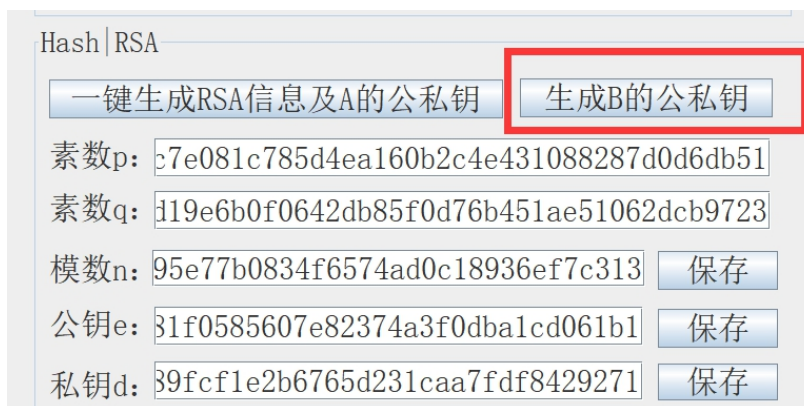
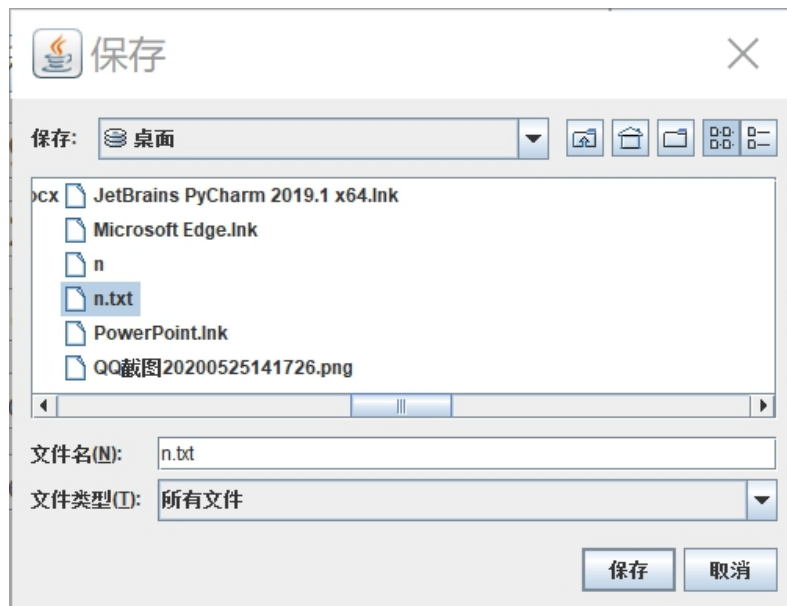
该接口函数为：

```
btnrsaa.addActionListener(new ActionListener() {
    public void actionPerformed(ActionEvent e) { }
});
```

```
btncb_1.addActionListener(new ActionListener() {
    public void actionPerformed(ActionEvent e) { }
});
```

//该模块为非对称加密模块，两个接口均调用自写的RSA加密函数RSA.RSA;，输出为素数 $p$ 、 $q$ ，模数 $n$ ，A、B的公钥 $e$ 和私钥 $d$ 。





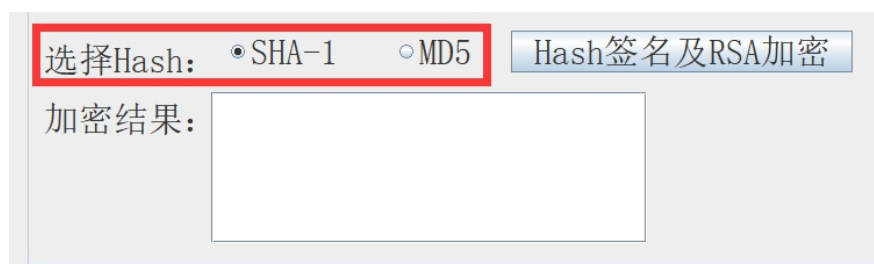
### 4.3 哈希签名

哈希签名模块分为SHA-1和MD5两种签名方式，选择签名算法后，点击右侧Hash签名及RSA加密按钮，将发送方A输入的明文进行哈希签名，用A的私钥对签名进行RSA非对称加密，加密结果展示在下方文本框内。

该接口函数为：

```
btnRsa.addActionListener(new ActionListener() {  
    public void actionPerformed(ActionEvent e) { }  
});
```

//该接口调用自写的哈希函数包，SHA1和MD5函数HASH.SHA1、HASH.MD5，输入为明文字符串，输出为加密后的十六进制字符串



### 4.4 对称加密

对称加密分为DES和AES两种加密方式，其密钥K也可选择，选择对称加密算法后，点击加密按钮，实现用B的公钥加密密钥K，并将之前对称加密后的签名结果用DES/AES算法进行加密，将二者结果组合起来，结果展示在下方文本框内。

该接口函数为：

```
btnrsa.addActionListener(new ActionListener() {  
    public void actionPerformed(ActionEvent e) {}  
});
```

//该接口调用自写的DES/AES加密函数，DES.DESUtil, AES.AES, AES.word，输入为十六进制字符串，输出分为两种。若该模块选择了DES加密，则输出为DES加密的二进制的字符串与十六进制字符串的组合；若该模块选择了AES加密，则输出结果为十六进制的字符串。



DES/AES/RSA

选择对称加密算法及密钥: -----请选择-----

对称加密&RSA加密对称密

加密结果:

DES及密钥: e53b7a65a

AES及密钥: 000120017

DES/AES/RSA

选择对称加密算法及密钥: DES及密钥: e53b...

对称加密&RSA加密对称密

加密结果:

001100011010100001011010110010001010

#### 4.5 发送密文

点击最下方的发送至用户B按钮，将加密结果通过安全信道发送给用户B。  
该接口函数为：

```
btnb.addActionListener(new ActionListener() {
    public void actionPerformed(ActionEvent e) {}
});
```

//该接口将最终的加密信息发送给用户B

DES/AES/RSA

选择对称加密算法及密钥: DES及密钥: e53b...

对称加密&RSA加密对称密

加密结果:

001100011010100001011010110010001010

发送至用户B

#### 4.6 接收密文

用户B点击左侧接受密文按钮，密文将显示在文本框内

该接口函数为：

```
button_1.addActionListener(new ActionListener() {  
    public void actionPerformed(ActionEvent e) {}  
});
```

//该接口输出接收的字符串密文



## 4.7 非对称加密的解密

逐个导入之前保存的n，A的公钥，B的公钥和B的私钥，选择RSA解密按钮，解密出对称密钥K。

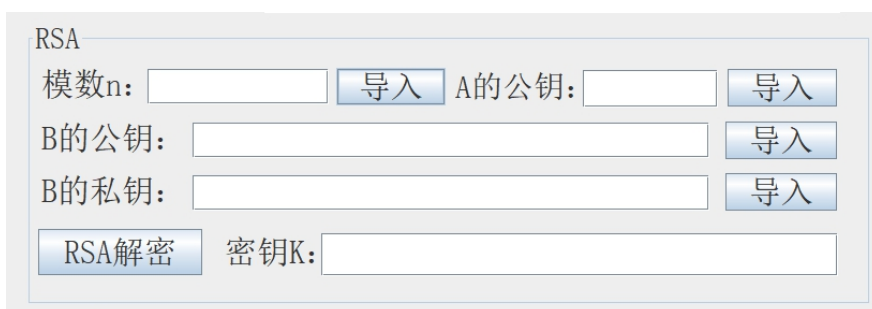
该接口函数为：

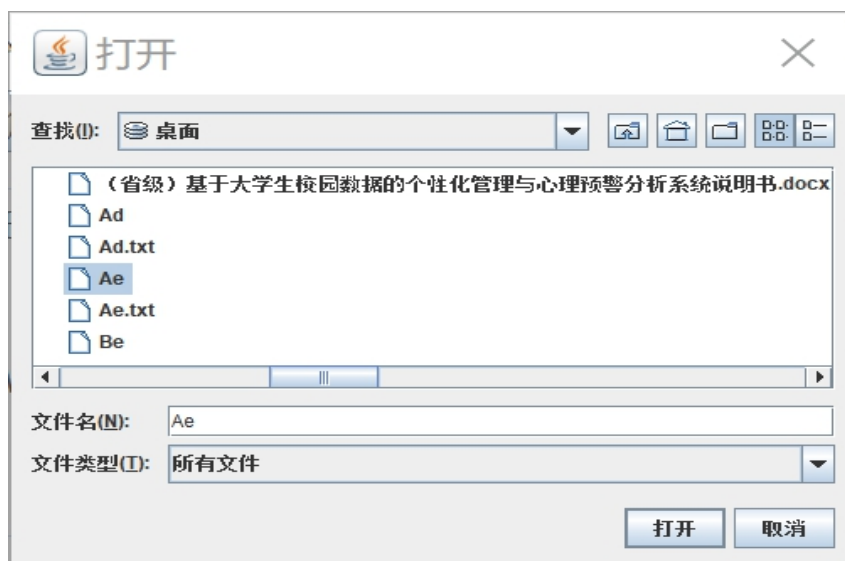
```
btnNewButton_4.addActionListener(new ActionListener() {  
    public void actionPerformed(ActionEvent e) {}  
});
```

//该接口为导入函数示例，点击按钮弹出文件选择框，该接口输入为文件，输出为文件内的十六进制的字符串信息，如模数n、A的公钥、B的公私钥。

```
btnRsak.addActionListener(new ActionListener() {  
    public void actionPerformed(ActionEvent e) {}  
});
```

//该接口为RSA解密函数，调用RSA.RSA函数，输入为十六进制字符串的模数n、A的公钥、B的公私钥信息，输出为十六进制字符串的密钥K





RSA

模数n:   A的公钥:

B的公钥:

B的私钥:

密钥K:

#### 4.8 对称加密的解密

选择与之前同样的对称加密算法，点击右侧按钮，下方文本框内展示解密出的明文及RSA (Hash (M))。

该接口函数为：

```
btnDesres.addActionListener(new ActionListener() {
    public void actionPerformed(ActionEvent e) {}
});
```

//该接口为DES/AES解密函数，调用自写的DES、AES解密函数，输入为十六进制字符串密文，输出为字符串明文与十六进制的字符串密文RSA (Hash (M))

DES/AES

☒ DES ☐ AES

明文M:  RSA (Hash (M)) :

## 4.9 再次哈希签名

选择与之前相同的哈希签名算法，点击左侧签名按钮，左下方文本框展示对明文进行哈希签名的结果；点击右侧RSA解密按钮，右下方文本框展示解密出发送方发送的Hash (M)。

该接口函数为：

```
btnm.addActionListener(new ActionListener() {
    public void actionPerformed(ActionEvent e) {}
});
```

//该接口为哈希签名模块，调用自写的SHA-1、MD5函数，输入为字符串明文，输出为字符串密文Hash (M)

```
btnNewButton_1.addActionListener(new ActionListener() {
    public void actionPerformed(ActionEvent e) {}
});
```

//该接口为RSA解密模块，调用自写的RSA解密函数，输入为十六进制字符串密文，输出为字符串密文Hash (M)

The interface consists of two main panels: 'Hash' and 'RSA'. The 'Hash' panel has radio buttons for 'SHA-1' (selected) and 'MD5', a '签名M' button, and a text field for 'Hash (M):' containing '85e0cc5d9d0abf0'. The 'RSA' panel has an 'RSA解密' button and an empty 'Hash (M):' text field. A '比较' button is located at the bottom of the interface.

#### 4.10 比较

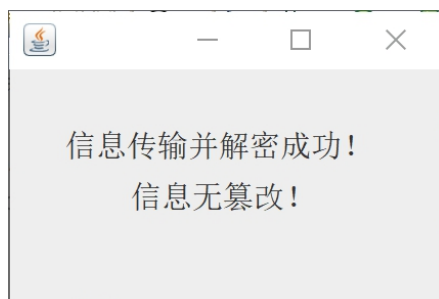
接受方将自己接收到的明文进行签名后的结果与发送方进行签名的结果进行比较，确认信息在发送过程是否被篡改，若比较结果相同，信息传输成功，若比较结果不同，信息传输失败。

该接口函数为：

```
button.addActionListener(new ActionListener() {
    public void actionPerformed(ActionEvent e) {}
});
```

//该接口为比较模块，输入为字符串密文，输出为成功或失败窗口。

This screenshot shows the same interface as before, but with the '比较' button highlighted by a red rectangle. The 'Hash (M):' field in the 'RSA' panel now contains the same value '85e0cc5d9d0abf0' as the 'Hash' panel.



# 五、界面设计

## 5.1 发送方A的界面

194588陶蕊

—□×

欢迎您，用户A!

输入

输入字符串:

或

选择文件:

Hash | RSA

一键生成RSA信息及A的公私钥

生成B的公私钥

素数p:

素数q:

模数n:

保存

公钥:

保存

私钥:

保存

选择Hash:

☐ SHA-1

☐ MD5

Hash签名及RSA加密

加密结果:

DES/AES/RSA

选择对称加密算法及密钥:

-----请选择-----

对称加密&RSA加密对称密钥

加密结果:

发送至用户B

发送方A的界面

5.2 接收方B的界面

194588陶蕊

—

□

×

欢迎您，用户B！

接收密文

RSA

模数n: 

导入

 A的公钥: 

导入

B的公钥: 

导入

B的私钥: 

导入

RSA解密

 密钥K:

DES/AES

☐ DES ☐ AES

DES/AES解密

明文M:  RSA (Hash (M)) :

Hash

☐ SHA-1 ☐ MD5

签名M

Hash (M) :

RSA

RSA解密

Hash (M) :

比较

接收方B的界面

六、数据存储方式

本系统中所用到的数据如模数n，A、B的公私钥均采用本地.txt文档存取方式。

