RSA 算法算法报告

实验目的

熟悉 RSA 算法加密和解密的过程，并且通过编程实现算法，达到对 txt 文件内容加解密的目的。

实验环境

JAVA
Eclipse
Windows
JDK 1.8

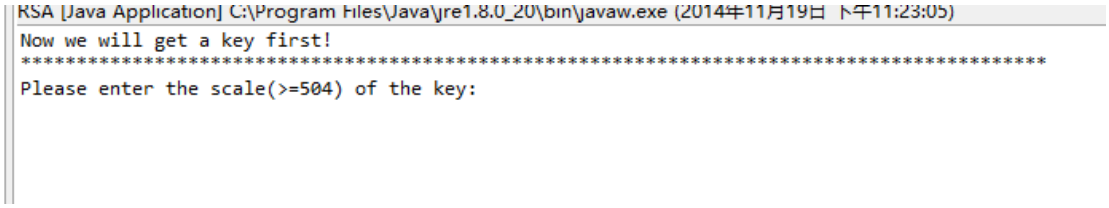
实验内容

1. 输入密钥规模，程序自动生成两个大素数，并显示；
2. 输入一个公钥，程序自动生成一个私钥，并显示；
3. 可以对指定的 txt 文档进行加密或者脱密处理，并生成对应的密文或者明文 txt 文档。

实验步骤

下面将通过运行过程来说明实验的步骤：
本次实验的源码主要包括两个类，分别是：RSA.class, RSA_key.class.其中 RSA.class 是主类，RSA_key 包括了公钥和私钥的信息。
注意，本次所有的 txt 文档都必须存在 D 盘根目录下才能正常读取。创建的新文档也会存在 D 盘根目录下。并且输入文件名直接输入文件名，不需要加后缀.txt
运行过程：

运行程序，出现如下信息：



我们需要先产生一个 key ,输入一个数字代表秘钥规模，要求大于等于 504 位，我们输入 504

```
Now we will get a key first!
********************************************************************************************

Please enter the scale(>=504) of the key:504
********************************************************************************************

Recommended encode key: 156413433212901052405466699199084280953724099346082716060894394633633192121473138509016323429963587942840781592293042
Enter the encode key you like:
```

要求输入一个公钥，由于数字太大，用户自己选择有困难，会调用一个 Findkey 的方法去自动寻找一个适合的公钥作为推荐公钥给用户，我们就选取这个公钥

```
********************************************************************************************
The first prime number is 781278923456655133
The second prime number is 42532177822786666270785456972964004789002746175782201699790439904621980702961898617799344109037353552508345031139788
The public key N is 33229494101653788803243983704412304647773562787392173218848106910629913212510512974783959502858773801699162942930016423951
LN is 33229494101653788760711805881625638376988105814428168429845360734847711512720073070161978799896875183899818833892662871407365668914628282
The public encode key is 156413433212901052405466699199084280953724099346082716060894394633633192121473138509016323429963587942840781592293042
The private decode key is 11074548419020479807203136884123921516753010844120005635092641357732872053537065831963175207116613528107143726942261
********************************************************************************************
--------------------------------------------------------------------------------
```

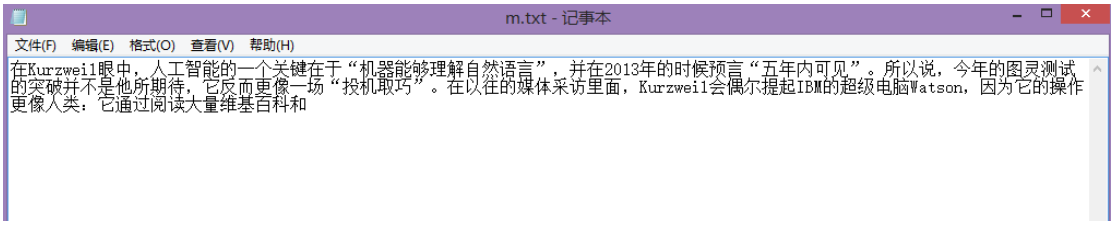
程序输入了和该秘钥有关的所有信息，包括私钥，N 等信息，还有两个素数的信息，由于数字太大，就不一一指出了，下面进入控制界面：

```
**********************************************************
What do you want to do?Please enter the number of the exact operation
1.Encode some file
2.Decode some file
3.Exit
4.Get a new key
Your choice:
```

1 号功能加密指定文件并输出，2 号功能解密指定文件并输出，3 号功能退出，4 号功能产生一个新的秘钥。我们输入 1，进行加密

```
**********************************************************
All files must be put in directory: D:/
Enter the file name to be encrypted without .txt
```

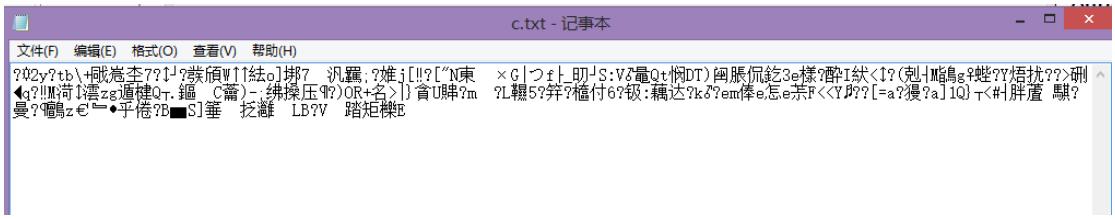
要求输入两个文件名，第一个是将要加密的文件名，必须存在，第二个是输出密文的文件名，如果不存在，将会自动创建一个。我们输入 m 作为明文文档，c 作为输出密文的文档，明文文档信息为：



下面输入文档名：

```
**********************************************************
All files must be put in directory: D:/
Enter the file name to be encrypted without .txt
m
Enter the file name to output the encrypted data without .txt
c
Encryption done!
**********************************************************
```

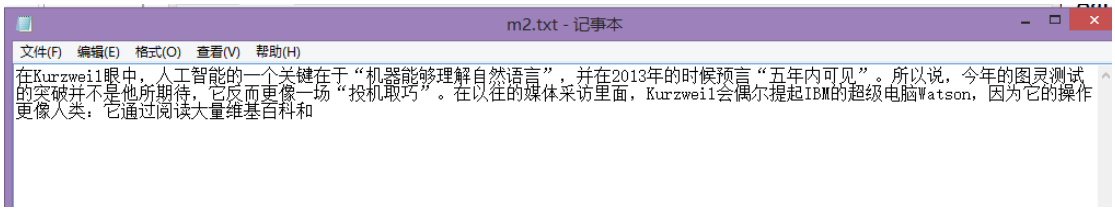
查看生成的 c.txt 内容



接下来重新解密该文档，在 m2 中输出明文信息

```
What do you want to do?Please enter the number of the exact operation
1.Encode some file
2.Decode some file
3.Exit
4.Get a new key
Your choice:2
******************************************************
All files must be put in directory: D:/
Enter the file name to be decrypted without .txt
c
Enter the file name to output the decrypted data without .txt
m2
Decryption done!
******************************************************
```

查看新生成 m2.txt 内容



对照发现和明文完全相同。注意，由于大数字生成函数的缺陷，会有一定几率出现解密的时候部分内容为乱码，这个时候就需要选取一个新的秘钥重新加减密。公布秘钥之前，公布者需要首先测试自己的生成秘钥能否使用。

具体实现方法在源码中有详细的注释。

实验的难点

本次实验主要需要借助 JAVA 中一个很重要的包，BigInteger，主要用以里和大数有关的问题。主要包括自动求逆元，产生一定位数的随机数，并以一定概率产生大素数，判断某个大数是否是素数等等。

主要难点是对于接口函数的不熟悉，不太清楚内部的原理而产生很多的问题，比如随机产生一个大素数，不一定就是素数，而只是一定几率。后来修改参数使这个概率增大，但是依然会出现合数，使加减密出错，这个时候需要重新加密。掌握了理论的算法实现过程后，在实际编写程序中，还会遇到各种不同的问题，不过大都是因为粗心没有考虑周全，没有其他的大问题。