key generation program

symmetric.key   binary o: class BufferedOutputStream

write(byte[ ] b, int off, int len)

Example code:

```
BufferedOutputStream symKeyFile = new BufferedOutputStream(
             new FileOutputStream("symmetric.key"));
//assuming that the 16-character user input has been written to skUserInput
//, which is a String object.
byte[] symKey = skUserInput.getBytes("UTF-8");
symKeyFile.write(symKey, 0, symKey.length);
...
symKeyFile.close();
```

XPublic.key	write
XPrivate.key	write
YPublic.key	write
YPrivate.key	write

object Output Stream

See RSAConfidentiality.java

How to read from a message file containing M piece by piece?

Class BufferedInputStream

public int read(byte[] b, int off, int len)

Class Cipher

byte[]   doFinal(byte[] input, int inputOffset, int inputLen)

```
//assuming that usrInput is a String containing user input
//regarding the file name of the message file.
BufferedInputStream msgFile = new BufferedInputStream(
             new FileInputStream(usrInput));

//create a byte array whose size is BLOCK_SIZE (option1: 117;
// options 2&3: 16KB=16*1024)
byte[] plaintext[BLOCK_SIZE];
//assuming this array is named plaintext[ ]
//int numBytesRead;
numBytesRead = msgFile.read(plaintext, 0, plaintext.length);

if (numBytesRead <= 0)
    break;

//if numBytesRead is less than plaintext.length but still positive,
//still encrypt the plaintext[ ] but only encrypt numBytesRead bytes.
//but this loop must terminate after completing doFinal() and maybe more
// in the current iteration
...
doFinal(plaintext, 0, numBytesRead);
...
```