**Project 2**

CS-4780

Howard Nguyen, Andres Vicente, Antonio Vazquez Bravo

**Generating Keys (RSAKeyGen.java)**

The java file can generate RSA keys by two methods, by providing the number of bits for a randomly generated key, or by providing two primes and the e value to generate a specific key. Keys will be saved on two text files, public\_key.txt and private\_key.txt, but can also be used as an instance variable when calling the class’s constructor. When using an RSAKeyGen instance, both keys are arrays of two values with e or d value as the first and the n value as the second. Both can be retrieved with their getter functions.

**Steps (running the file with arguments)**

1. In terminal run the file as one of the following
   1. For specific generated key: **java RSAKeyGen <p> <q> <e>**
      1. Where <p>, <q>, and <e> are the primes and values to enter.
   2. For random generation: **java RSAKeyGen <number of bits>**
      1. Where the argument passed is the amount of bits the key will have.
2. Two files will be created, public\_key.txt and private\_key.txt, with each file having both values on a seperate line.

**Steps (using an instance)**

1. Import RSAKeyGen
2. Create a new instance of RSAKeyGen class. Providing an integer will generate a random key, while providing 3 number strings will generate that specific key.
3. Both public and private keys can be read with getter functions as arrays with their two values.

**File Encryption(RSAKeyGen.java)**

The program will encrypt an arbitrary file by providing two command-line arguments, a file containing a message to be encrypted, and another file containing the RSA public key. The program reads the message, removes non-alphabetic characters, then encrypts the message in blocks of three characters by using the provided public key. The encrypted blocks are written to an output file named test.enc and includes error handling, and outputs success message upon encrypting message.

**Steps**

1. In terminal run with two arguments:

**java RSAKeyGen <path of the file to encrypt> <path of the file with the key>**

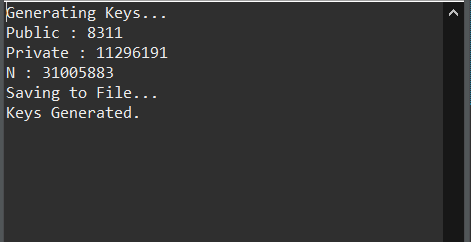
1. A new file named, test.enc, will be created with the encrypted contents.

**File Decryption**

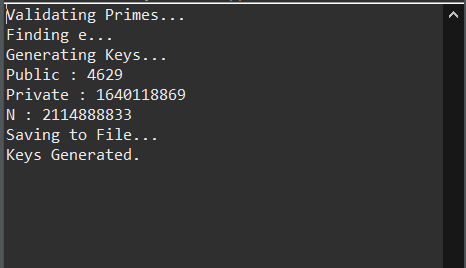
Decrypting files encoded with RSAEncrypt is the same process as encryption. Just run the RSADecrypt file providing the encrypted file path and the decoding key file path as arguments similar to encryption. You can refer to the steps in file encryption for detail. The program will create a file named, test.dec, with the decrypted contents in it.

**Screenshots**

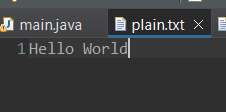
**Output of key generation with p=6551 q=4733 e=8311**

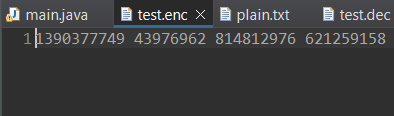
****

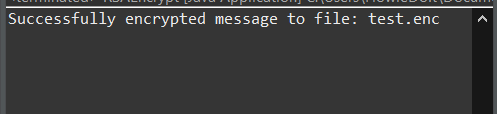
**Output of random key generation**

****

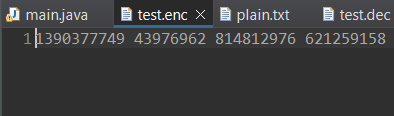
**File Encryption**

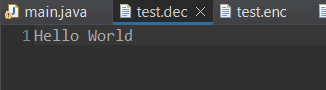
****

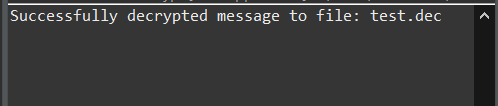
****

****

**File Decrypt**

****

****

****