SurajSG23 Create RSA 6d0d58d · now

62 lines (49 loc) · 3.76 KB

Code Blame Raw

```
1   import java.util.*;
2   import java.math.*;
3   import java.nio.charset.*;
4
5   public class RSA{
6           public static void main(String[] args){
7                   BigInteger p,q,N,phi,e,d;
8
9                   p = BigInteger.probablePrime(1024,new Random());
10                  q = BigInteger.probablePrime(1024,new Random());
11                  N = p.multiply(q);
13                  e = BigInteger.probablePrime(512,new Random());
14
15                  //e and phi should be co-prime (gcd of e and phi = 1 ) & 0 < e <
16                  while( phi.gcd(e).compareTo(BigInteger.ONE)>0 && e.compareTo(phi
17                          e = e.add(BigInteger.ONE);
18                  }
19
20                  d = e.modInverse(phi);
21
22                  System.out.println("Prime number p: "+ p);
23                  System.out.println("Prime number q: "+ q);
24                  System.out.println("Public key is: "+ e);
25                  System.out.println("Private key is: "+ d);
26
27                  Scanner sc = new Scanner(System.in);
28                  System.out.print("Enter the plain text: ");
29                  String testString = sc.nextLine();
30                  System.out.println("Encrypting String: "+ testString);
31
32                  byte[] encrypted = new BigInteger(testString.getBytes()).modPow(
33                  byte[] decrypted = new BigInteger(encrypted).modPow(d,N).toByteA
34
35
36                  System.out.print("Encrypted Bytes: ");
```

```
                    for(int i=0; i<encrypted.length; i++){
                            System.out.print(encrypted[i]);
                    }
                    System.out.println();

                    System.out.print("Decrypted Bytes: ");
                    for(int i=0; i<decrypted.length; i++){
                            System.out.print(decrypted[i]);
                    }
                    System.out.println();

                    System.out.println("Decrypted String: " + new String(decrypted, 
            }
    }

    //Output

    Prime number p: 154288893773697694135804882599026753843801559426943509387350637
    Prime number q: 179111106009266559045579148761777903274690701821600981757018012
    Public key is: 114650930183927022889441941528952760459488054175062927864078162025
    Private key is: 174285662842972426969425000877867829058347197268152641796254224
    Enter the plain text: Hello i am Suraj
    Encrypting String: Hello i am Suraj
    Encrypted Bytes: 62-365666-119-20-6897-47-8670-66-10632-75-20-28116-65-296413124
    Decrypted Bytes: 72101108108111321053297109328311711497106
    Decrypted String: Hello i am Suraj
```