CS 415 Project 2 Spring 2017

Due: March 19, 2017

`This HW will be done in pairs. Each team will have a single submission.No sharing of code with other teams please. This HW will weigh 4 points.`

The goal of this project is to implement the algorithms presented in class for the following problems: modular exponentiation, modular inverse (by using Extended Euclid's algorithm), primaility testing, RSA key generation, Encryption and Decryption of a message using the RSA algorithm.

PROBLEM 1:

The goal of this problem is to implement the faster primality test (we called **primality3** in the quiz and the mid-term): Given as input an integer N and confidence parameter k, first test of N is divisible by 2, 3, 5 or 7. If it is divisible by any of these numbers output('no'); else call **primality2** with N and k as inputs. (This in turn call **primality** algorithm that randomly chooses $1 < a < N$ and tests if $a^{N-1} \equiv 1 \pmod{N}$ and repeats it k times.)

PROBLEM 2:

Given an integer n ($k \leq 50$) and k, generate a random prime number with n bits. Implement a solution as follows: generate a random $k-2$ bit vector v and add 1 to both ends (as the first and the last bit) to create a k bit integer. The reason for the first bit to be 0 is that we want the number to be odd. The last bit should be 1 since we want no leading 0's. Then, call **primality3** algorithm you in implemented in Problem 1 with v and k as inputs. (k is the second parameter in primality2 which is the number of times Fermat test is repeated.) Repeat calling primaility3 until it outputs 'yes'. At this point, you have found a prime number (with high probability).

PROBLEM 3:

Given integers n and k, call the algorithm for Problem 2 (with n and k as inputs) to generate two primes p and q with n bits each, and use them to generate the encryption keys N and E and the decryption key D. After computing p and q, compute $N = p \times q$. Then, find a random k-bit integer E such that $\gcd(E, (p-1)(q-1)) = 1$. Next, find D such that $DE \equiv 1$ $(mod\ N)$ using extended Euclid's algorithm. Output N, E and D.

PROBLEM 4:

Given integers N, E, D and M where (N, E) form the RSA encyrption keys and D the decryption key, encrypt the plain-text message M and decrypt it and print both. (If every thing works fine, the final decrypted message must be the same as the plain-text M.)

When your program is executed, it should allow the option for the user to test Problem1 (option 1), Problem 2 (optin 2), Problem 3 (option 3) or quit (option 4). After the user enters the option as the number in $\{1, 2, 3, 4\}$, the program asks for the appropriate inputs for that problem, solves it and outputs the result. It then repeats the query above until the user chooses to quit.

As in Project 1, all the computations will be done in binary using the array representation, but the user input will be in decimal. Use the Code from Project 1 for converting from binary to decimal and from decimal to binary

Your submission shall include the source code, compiled code and instructions on executing the code. The submission details will be posted via piazza.