**CS462 Networking Security**

**Project 3 – Public Key Cryptography**

**Tasks**

In this project, you are asked to implement a basic version of RSA public-key cryptography and test its use in encryption and digital signature.

**Phase I (70 points): Team Programming Work.**

1. Create a function/method to **find a prime number** that is closest to a given number that is provided as the parameter. Use Miller Rabin’s algorithm to do the primality test during the search of a prime number, remember to test at least 10 different values of *a* before claiming a number is likely prime. Note that you need to implement the Miller Rabin’s algorithm ( do not call an existing function/method)
2. Create a function/method to **find the private key** based on given values of e and ⱷ(n) using extended Euclidean algorithm (Refer to textbook Page 97-99, let b = e, a= ⱷ(n), then the final value y is the private key that you want) . Note that you need to implement the Euclidean algorithm (do not call an existing function/method)
3. Create a generic **RSA encryption/decryption function/method** by using the exponentiation modular operation (Refer to textbook Page 267-269, or use the recursive implementation we discussed in class). This function/method can be used for either encryption or decryption depends on the arguments used.
4. Create a main function/method to:
   1. Generate two large prime numbers p and q with each number being at least 80 digits long.
   2. Ask user to pick up a value of public key e (e.g. 17, 2^16-1). Call the private key computation function/method using the values of e, p, q obtained to find the private key d.
   3. Ask user to input a two digit number (e.g. 98), call the encryption function/method to encrypt the small number. Then call the decryption function to recover the plaintext small number.

***Note:***

1. ***Please form a group by Tuesday 11/14 /17 and email me your group list.***
2. ***Each member is responsible for at least one of tasks (1), (2), and (3).***
3. ***The team should work together on the main function/method and test together to make sure it can encrypt and decrypt correctly.***
4. ***You need to use* big integer class *in Java/C++ to complement the above functions/methods. For those of you who use C++, you need to figure out ways to use the big integer library in the first week. Do some research and please feel free to discuss the environment configuration issues on piazza.***

**Phase II (30 points): Individual work on processing, tests, and online discussions**

1. Everyone is required to ***generate a public key (value of e and n) and post it*** as a thread on D2L discussion board under the title “Project 3 Decryption Game”. ***Keep your private key*** in a safe place so that you can retrieve it later.
2. Everyone is required to post two ciphertexts under two different threads of others. To do this
   1. You need to construct a short sentence that is around 10 letters.
   2. Convert each letter into a decimal number by checking the ASCII table.
   3. Encrypt the message (letter by letter) using your RSA encryption algorithm and the public key provided by others (under other’s threads). Note that since the ciphertext for each number might be huge, separate them into multiple lines so that others can identify easily.
   4. To avoid uneven posts, do not post reply to threads that already have 3 ciphertexts posted.
3. For each ciphertext one got under your original thread, you should reply each of them by showing the plaintext recovered by you using your RSA decryption program. To recover the plaintext, you should use the private key you kept and call the RSA decryption function/method. Then you need to convert them back to a string sentence. Check whether the sentence makes sense. You can double check with the author who posted the ciphertext.

***Note:***

1. ***Each member use the programs obtained as team effort in Phase I.***
2. ***You need to write additional functions/methods to enable you to work on a whole sentence in encryption and decryption. Use of file input/output is encouraged.***
3. ***If in your test you found problems with your team program (either of the three major functions/methods) , your group should work together to fix the problem.***

**Submission:**

What you need to submit on D2L:

1. The Java/C++ programs for Phase I tasks (everyone should submit one copy of the team work, make sure your author name appears in the block comments of the function/method you wrote).
2. A peer evaluation form that evaluates the contributions of you and your collaborators.
3. The Java/C++ programs for Phase II tasks
4. Your posts on the discussion board for Project 3 Decryption Game thread.

Put all of your files in a zipped folder and upload it to D2L Project 3 dropbox.