3/27/14

8. Homework

Programming portion due **Thursday 4/3/14** at 11:55pm on Blackboard.

Please submit one `.cpp` file on Blackboard.
In order to receive any credit for the programming portions, you are required to thoroughly comment and test your code.

1. **Public Key Cryptography (20 points)**
   If you want to send a message via a non-secure channel, you may want to encrypt your message such that an eavesdropper cannot easily listen in on your communication. This is particularly important in transactions conducted over the internet, such as banking or shopping. An encryption method that has already been used in Roman times is the *shift cipher*: Consider a string that you want to encrypt, say the word "Hello". Look up the ASCII code for each character, this gives the sequence 72 101 108 108 111. Then shift each character-code by a fixed number, and consider this to be the encrypted message. For a shift of 5 this would yield 77 106 113 113 116 or as a string "Mjqqt". Now one can send the encrypted message "Mjqqt" over an insecure channel, and any eavesdropper will have a hard time reading the message. Decoding is of course done by shifting each character by -5. But somehow, the shift amount of 5 also needs to be sent to the receiver of the message as well.

   Shift ciphers are in general not very secure, but let's not worry about that right now. We can make them a little more secure by encrypting the shift amount itself with something called public-key cryptography. You've probably heard of RSA; that's a public-key cryptography system that is based on the fact that factorizing large primes is hard. We are going to implement a simple version of RSA in this homework assignment.

   Download the file `hw8.cpp` and add C++ code to it to address the following questions. Split the function and class declarations from the implementations.

   (a) (5 points) Let's start with a simple shift cipher. Implement and test a function `char* shiftString(int shift, char* input)` that takes as input a shift amount and an input string, and that returns a (pointer to) a result string. The storage for the result string has to be allocated inside the function. The result string should consist of the characters in the input string each shifted by `shift`. Since some of the ASCII characters are special characters that don't print well, we are going to limit ourselves to characters with ASCII codes between 32 and 126, and the shift will be performed cyclically. A function `char shiftChar(int shift, char c)` that shifts a single character has been provided. Remember, strings in C/C++ end with the '\0' character.
   What does "Hello World" look like when it is shifted by 10? And can you shift the result back by -10 and obtain "Hello World"?

Flip over to back page ⟹

(b) (15 points) In RSA cryptography, every user has a public key (consisting of two numbers e and n) and a private key (consisting of two numbers d and n). The public key is stored in a public database which is accessible to everyone; similar to the notion of a phone book. Every user also has a secret private key that nobody else knows. A message is simply a positive integer (it will be the shift amount a, in our case).

Let's say the message is the shift amount a. Then the encrypted shift amount is $a^e \bmod n$; note that we only needed to know a and the public key (e, n) for the encryption. Decryption of a "mangled" shift amount b is done by computing $b^d \bmod n$; note we used the private key (d, n) for that. So, if you want to send a message to a user "Alice", then you just look up Alice's public key, encrypt the message, and send it to her. Alice is the only one who can decrypt the message with her secret private key.

We will combine RSA with the shift cipher as follows: For an input message "Hello" and the (arbitrarily picked) shift amount 5, we will send the message "Mjqqt" to Alice together with the encrypted shift amount (which might be 85 for example).

- Implement a class `User` that stores the private key (d, n). In its constructor it should use the provided function `next_n_e_d(int* n, int* e, int* d)` to initialize the public and private keys for this user, and then store the public key in `PublicKeyDB`. It should contain a private method `int decryptNumber(int shift_encrypted)` which takes an encrypted shift amount as input, and uses the provided function `pow_mod` to decrypt the amount. It should also contain a function `void receiveMessage(int shift_encrypted, char* shifted_message)`, which decrypts the shift amount and shifts the `shifted_message` back by the negation of the shift amount. This particular function should print out status updates of the decryption of the message and print the cleartext.
- Implement a class `PublicKeyDB` that serves as a lookup-table for the public keys. It should be able to store for each user the public keys e and n as well as the address of the `User` object; you can use a struct or a class to group these three together and store them in an array. `PublicKeyDB` should have a constructor that initializes this container class, a `setKey(...)` method that takes the public key e and n as well as the address of the `User`, and stores them together in the array. There should also be a private method `getPublicKey(User *user)` that returns the public key (e, n) for a given `User` address. Finally, there should be a method `int encryptNumber(User* recipient, int shift)` that looks up the public key for a user and returns the encrypted shift amount.
- In the `main` function, perform the following tests: Create at least two `User` objects "Alice" and "Bob". Then pick some message and some shift amount, encrypt them for Alice and send this meta message to her (using her `receiveMessage` method), and see if she can decrypt it. Pick a second message and a second shift amount, encrypt it and send it to Bob, and see if he can decrypt it. Can Alice decrypt the message that was meant for Bob?
- In the `main` method, declare Alice as the very first `User` object. Can Alice decrypt the message

  `^ro*~o}~*q|kno}*k|o*oxmynon*sx*K]MSS8`

  with encrypted shift amount 60?