Assignment 3

Due: 11:59pm June 21 (Sunday)

This assignment is done individually.

You can choose ONE of the following assignments:

Choice 1 (SSL, 10 points extra credits):

This assignment can be done using C/C++/Java/Python.

You will implement a client and a server using **Secure Socket Layer (SSL)**. SSL enables to establish a secure connection between the server and the client. Upon connection, the client prompts the user to enter his/her ID and password. After the user enters the ID and the password, the client sends the ID and the password to the server through SSL connection.

The server maintains a file ***password*** which has the following format:

alice 12345
bob 67890

Where alice and bob are IDs, and 12345 and 67890 are the corresponding passwords.

After the server receives the ID and the password, the server verifies the ID and the password. If the ID and the password are correct, then the server sends "correct ID and password" to the client and terminates. Otherwise, the server sends "incorrect ID and password" to the client and terminates.

In order to establish a connection between the client and the server. The client needs to specify the server's domain name (e.g. remote.cs.binghamton.edu) and port number. The port number is used to identify the server process, and is specified as a number between 1024 and 65535. In this assignment, you will hard-code the server's port number in both the client and the server.

The client has at least one argument <server_domain>, which specifies the domain of the server. You can add other arguments to the client and the server if needed.

In addition, you need to generate a public key certificate in order to establish the ssh connection. For example, in C, you can use the following command to generate certificate.

```
openssl req -x509 -newkey rsa:4096 -keyout key.pem -out cert.pem -days 365
```

If you use java, you can use keytool to generate the certificate.

Note:

You can use any code available on the web for SSL socket programming. However, you must write your own code for the rest part of the assignment (e.g. enter and verify ID and password, open/read files). You should also generate the certificate by yourself. Please use your name to generate the certificate (other information can be forged).

If you use C, please use remote.cs.binghamton.edu. The openssl installed on bingsuns may not work. To compile your C program, please use the following command:

```
gcc -Wall -w -o sslcli sslcli.c -I/usr/local/ssl/include/ -L/usr/local/ssl/lib -lssl -lcrypto
gcc -Wall -w -o sslserv sslserv.c -I/usr/local/ssl/include/ -L/usr/local/ssl/lib -lssl -lcrypto
```

Submission guideline

- Create a directory with a unique name (e.g. p3-[userid]), which contains the source code, the certificate, and a README file.
- **README** file (text file, please do not submit a .doc file) contains
  - Your name and email address.
  - Whether your code was tested on bingsuns or remote.cs.
  - How to compile and execute your program.
  - (Optional) Briefly describe your algorithm or anything special about your submission.

-Tar the contents of this directory using the following command.

tar –cvf p3-[userid].tar p3-[userid]

E.g. tar -cvf p3-pyang.tar p3-pyang/

- Upload the tared file you create above to mycourses.

Choice II: (RSA Implementation)

The assignment can be done using C/C++/Java/Python.

In this assignment, you will write a program to generate public/private key using the RSA algorithm, given below. Your program has two arguments p and q, which are two prime numbers.

1.Compute n=p*q and ø(n)=(p-1)(q-1), print the value of n and ø(n) using the following format:

n= <value of n>
phi(n)= <value of ø(n)>

2. Select at random the encryption key e such that 1<e<ø(n) and gcd(e,ø(n)) = 1, print the value of e using the following format

e= <value of e>

3. Solve the following equation to find decryption key d (0<=d<=n): e*d mod ø(n) = 1, and print the value of d using the following format

d= <value of d>

Execution (e.g. C):

./rsa 11 13

Submission guideline:

- Create a directory with a unique name (e.g. p3-[userid]), which contains the source code and a README file.
- **README** file (text file, please do not submit a .doc file) contains
  - Your name and email address.
  - Whether your code was tested on bingsuns or remote.cs.

- How to compile and execute your program.
- (Optional) Briefly describe your algorithm or anything special about your submission that the TA should take note of.

-Tar the contents of this directory using the following command.

tar –cvf p3-[userid].tar p3-[userid]

E.g. tar -cvf p3-pyang.tar p3-pyang/

- Upload the tared file you create above to mycourses.

Choice III: PGP

Download a PGP software that supports **confidentiality** AND **digital signature**, and show how to use PGP to provide confidentiality and digital signature. You can choose any email client and any PGP software.

Submission guideline:

Please record a video that shows how to use PGP to provide confidentiality and digital signature, upload the video to google drive, and email me (pyang@binghamton.edu) a link to the video.

Academic Honesty:

All students should follow Student Academic Honesty Code (**if you have not already read it, please read it carefully**). All forms of cheating will be treated with utmost seriousness. You may discuss the problems with other students, however, you must write your OWN codes and solutions. Discussing solutions to the problem is NOT acceptable. Copying an assignment from another student or allowing other students to copy your work may lead to an 0 in the assignment or an F in the course. Moss will be used to detect plagiarism in programming assignments. You need ensure that your code and documentation are protected and not accessible to other students. Use **chmod 700** command to change the permissions of your working directories before you start working on the assignments. If you have any questions about whether an act of collaboration may be treated as academic dishonesty, please consult the instructor before you collaborate.