Assignment 3

Due: 11:59pm March 29 (Thursday)

This assignment is done individually or by a group of 2.

You will implement a client and a server using **Secure Socket Layer (SSL)**. 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 password to the server through SSL connection.

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

<user ID> <hashed password> <date and time when the password is stored>

The password can be hashed using SHA1 or MD5. You can use the existing implementation of SHA1 and MD5 (e.g. the implementation provided in java.security, openssl, etc.). **You will need to write a program to generate file *password*. Your program will be invoked as**

./gen-pass (C/C++)
java Gen-pass (Java)

When gen-pass/Gen-pass is invoked, it prompts the person who invokes gen-pass to enter each user's ID and password. Your program then computes the hash of the password, and saves ID, the hashed password, and the date and time when the password is saved to file ***password***. Your program should also check whether the ID is already in the file. If so, your program displays "the ID already exists".

The SSL server is invoked as

sslserv <*server_port*> (C/C++)

java SslServ <*server_port* > (Java)

<*server_port*> specifies the port number on which the server listens for the connection.

The SSL client is invoked as

sslcli <*server_domain*> <*server_port*> (C/C++)

java SslCli <*server_domain*> <*server_port*> (Java)

<*server_domain*> specifies the domain name of the server, i.e., bingsuns.binghamton.edu

When the client connects to the server, the server prompts the client to enter the ID and password. After the server receives the ID and the password, the server computes the hash of the password, and prints the ID, the password, and the hashed password. The server then compares that hashed password against the password stored in the password file. If the two passwords match, the server sends a string "OK" to the client and the client prints "the password is correct" and terminates; otherwise, the client prints "the password is incorrect" and terminates.

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

Submission guideline

- Create a directory with a unique name (e.g. p3-[userid]), which contains the source codes, a makefile, and a README file.
- **README** file (text file, please do not submit a .doc file) contains
  - The name and email address of your group members.
  - Whether your code was tested on bingsuns.
  - How to 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.

Grading guideline

- Correct execution format: 2'
- Readme: 2'
- Makefile: 6'
- Correct implementation of gen-pass/Gen-pass: 35'
- Correct implementation of SSL server and client: 55'

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 students in other groups, however, you must write your OWN codes and solutions. Discussing solutions to the problem is NOT acceptable. Copying an assignment from students of another group or allowing students from another group to copy your work may lead to the following:

1. **Report to the department and school**
2. **0 in the assignment or F in this 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.