

Programming Assignment 3

1 Aim

The aim of this project is to get you cursory familiarity with the Java SSL API. This will enable you to tackle more complicated projects in the future in which you might want to build the SSL server side, require client authentication, create, issue and sign your own certificates for use by clients etc. Knowing about the callbacks that the Java SSL API makes at different points in the SSL handshake procedure will also help you understand how the API can be customized to suit your application. You should find a wealth of information on Java's implementation of crypto APIs at <http://java.sun.com/j2se/1.5.0/docs/index.html>. The Java security features are described at <http://java.sun.com/security/index.jsp>.

I will also put a chapter from the JDK 1.4 Tutorial by Gregory Travis that describes JSSE in much detail in the Science co-op. Not all of the chapter is necessary to do the project but it'll help you understand how JSSE works better.

2 Problem

Implement a simple Java based client that connects to a server using SSL and makes a request of the form

```
get <identifier>
```

where <identifier> is a sequence of non whitespace characters. The `get` line is terminated with a newline `'\n'`. In response, the server will return some data and close the connection. Your client should receive and print this data to `System.out` and exit. Your program should take two arguments, the *hostname* of the remote machine where the server is running and the *port* at which it's running. I should be able to test your program by running, for example

```
java SSLClient <host> <port>
```

I have the test server running on suna.comp.nus.edu.sg at port 7230 for you to test against. Make a guess as to what the server is returning! The (tentative) deadline for submission is midnight April 14.

3 Submission

You should submit a `jar`¹ file that includes all the sources (`.java` files) and a `README` file that describes the contents of all other files in your submission. If you've done something extra please mention it in the `README` file. Submit only java source files, not any `.class` files. Make sure that you package your source files so they'll `unjar` in the current directory and will *not* create any directories below. I should be able to compile your program by `unjarring` your submitted `jar` file and running `javac *.java` to create at least an `SSLClient.class`. Your program should be written in the default package.

Just mail me the `jar` file by the submission date.

¹ I don't know how to cope with `rar` files.