Client-Server Secure Messaging

The project contains 3 Java files, Client and Server which are their respective messaging processes, and Seclib which is a library containing security functionality that both Client and Server use.

Client and Server communicate over a socket connection on a previously agreed upon common port, by default it is 25000. String objects are sent and received by the client and Server. The Server is always up and waiting for a client to connect and send a message containing security parameters, these parameters must match the parameters assigned to the Server or the client will be rejected. Following establishment of security information the client and server now enter a text exchange. The client and server must communicate in alternating fashion starting with the client. The text exchange continues until one party breaks connection.

Confidentiality

If confidentiality is selected, every message between the clients, except for the initial security parameter message, will be encrypted by a key previously shared among the parties. The message is then decrypted upon arrival.

Integrity

If integrity is selected, every message between client will be accompanied by a SHA-256 hash of the message. The receiver will then verify the contents of the message by generating their own SHA-256 hash and comparing to the received one. The hash is sent in a separate message before the text in order to more strongly prevent interception and tampering.

Authenticity

If authenticity is selected, every message will be signed by the creator and verified by the receiver to be authentic using a digital signature.

Instructions:

1. compile and run the server with “javac Server.java” and “java Server”.
2. The Server will then prompt the user for security parameters, enter “Y” or “N” for yes or no to the proposed parameters.

```
C:\Users\Tal\Desktop\school\seng360\seng360\src [master +2 ~0 -0 !]> java Server
Server Started and listening to the port 25000
Does this session require confidentiality? Y/N

y

Does this session require integrity? Y/N

n

Does this session require authentication? Y/N

y
```

3. In a separate terminal window, compile and run the client with “javac Client.java” and “java Client”.
4. Similar to the Server, the Client will prompt with security parameters, enter “Y” or “N” as desired. These parameters must match the configuration of the server.

```
C:\Users\Tal\Desktop\school\seng360\seng360\src [master +3 ~0 -0 !]> java Client
Does this session require confidentiality? Y/N

y

Does this session require integrity? Y/N

n

Does this session require authentication? Y/N

y

Message sent to the server : SecurityParametersIncoming

Message received from the server : Request acknowledged
Message sent to the server : 101

Message received from the server : Security parameters accepted
No need to close socket because message received was Security parameters accepted
Message to server :
```

5. if the security configuration was accepted by the server the client is now ready to send messages. Enter any desired string to send securely to the server.
6. As the server reply with any desired string.
7. repeat steps 5 and 6 indefinitely.