CS 5490/6490: Network Security – Spring 2023
Programming Assignment 2
Due by 11:59 PM MDT on March 30th, 2023

Important: No cheating will be tolerated. No copying from the Internet is allowed.

Total Points for this Programming Assignment: 100

The goal of your program is to build your own simplified version of SSL, called *mySSL,* in Java or Python. Use client server sockets to create a TCP connection. Also, use the SSL record format for both your handshake and data phase messages. Your client server programs must do the following:

Handshake Phase (to begin with, use the SSL handshake in the class notes) (*45 points*)

- The client and the server authenticate each other using certificates. You need to create the certificates (self-signed) and include them in the mySSL messages.
- The client also informs the server what data encryption and integrity protection scheme to use (there is no negotiation). Pick your favorite integrity protection and encryption algorithms.
- The client and server also send encrypted nonces to each other encrypted with the other side's public key (the public keys are obtained from the certificates). These nonces are then *xored* to create a master secret.
- Compute a hash of all messages exchanged at both the client and server and exchange these hashes. Use keyed SHA-1 for computing the hash. The client appends the string CLIENT for computing its keyed hash and the server appends the string SERVER for computing its keyed hash. Verify the keyed hashes at the client and the server.
- Generate four keys (two each for encryption, authentication, in each direction of the communication between the client and the server) using this master secret. Pick your own key generation function (should be a function of the master secret). You can use a hash function, if you like.

Data Phase (*45 points*)

- Transfer a file, at least 50 Kbytes long, from the server to client.
- Decrypt the file at the client and do a *diff* of the original and the decrypted file to ensure that the secure file transfer was successful.

(*5 points* for comments in the code, *5 points* for error/exception handling in the code)

Use *opnessl* or any other security library of your choice in any form convenient to you to generate certificates and to extract public keys from certificates and also for keyed hash computation, encryption, and data integrity protection.

Include print commands in your code to show

1. a failed verification of keyed hashes (possibly due to corruption or changes in one of the handshake messages), and

2. a successful client-server mutual authentication, key establishment, and secure data transfer.

Submit your code along with the output files, and a readme file. The readme file should briefly explain how the code is organized. Your code should be well commented. You will be required to show a demo of your programs to the TA. You can use any computer/laptop to work on your programming assignment.

Note: Please use the broad guidelines stated above for completing your assignment. Specific implementation choices are left to you.

References for SSL Record Format: There are plenty of documents available online on SSL. Feel free to use them but do not copy any code. Examples include:

- https://www.cisco.com/c/en/us/support/docs/security-vpn/secure-socket-layer-ssl/116181-technote-product-00.html#anc2
- https://www.ibm.com/support/knowledgecenter/SSB23S_1.1.0.12/gtps7/s5rcd.html