**Project Description**

How it works

The program aims to create a secure connection between clients and a server. The program does so by implementing a simplified version of RSA encryption and AES encryption. After which, the program then authenticates the clients based on their agent connection code and a security question.

Agent

* The client takes the server ID and port number through which the connection will be made
* Then the socket is created and the connection made between the client and the server
* Four functions are defined. Firstly, the send(msg) function which takes any msg from the agent, encodes it and sends it to the server. Then the getConCode() fuction will accept the agent’s input of their connection code and return it. The getAnswer(question) function will display a question from the server to the agent, accept the agent’s answer then return the answer. Finally, the computeSessionKey(n) function will compute the session key for a given node.
* In the program the agent first sends a message “100 Hello” to the server and the server responds with their public key, then agent computes a session key. The message “103 Session Key” along with the computed session key ,which is RSA encrypted, is sent to the server. Then agent’s symmetric key is set up and the nonce, sent with the servers public key, is encrypted (AEs encryption)with the symmetric key and sent to the server. The agent receives a “200 ok” message if the encrypted data sent is correct (nonce).
* The program then uses getConnCode() to retrieve the agent’s code, encrypts it and sends this to the user using send function. The client is then expecting a response from the server in the form of a question. Which the client will only receive if the connection was correct. Afterward the client gives an encrypted answer to the question and if the answer is correct the receive a welcome message from the server which is decrypted and printed on the screen.

Server

* Firstly, the server socket is initialized then the sever address(including server IP and port number) is binded to the server socket.
* The clientHandler function is defined which accepts connection socket, address and state. In this function the server receives the “100 Hello” message from the agent and responds by sending its public key and nonce. The server then receives message “103 Session Key”, the encrypted symmetric key and the nonce which is encrypted using the symmetric key. The function will thw decrypt the nonce and check if it matches the nonce previously sent to the agent, if so a “200 OK” message is sent and the server awaits the agent connection code. If the nonces do not match then th connection is closed.
* Then the connection code of the agent is received and validated. If the code is invalid the connection is closed, otherwise the server send a security question to the client. The answer fron the client is the received and checked if the answer is incorrect the connection socket id closed, otherwise the server send a welcome msg to the client.

Verify

* Verify supports the server in its authentication process
* The program creates a list of all possible Agent A and Agent B codes. It uses random to choose a question and answer tuple from a list of tuples and the server uses this function to get the question it will ask the client and the response that its will check with the clients response.
* There is also a function that takes a connection code and compares it to all possible Agent A and Agent B codes and it returns the agent that the code corresponds to. If the code has no matches the function returns -1

Design Trade offs Made/Considered

* A possible design trade of is to have a message sent to the agent if the answer is incorrect (“Incorrect Answer”) to let the agent know that they gave the wrong answer
* Another possible design trade off considered was creating and error message whenever a non-prime number is entered in for P and Q on the server side.
* A design trade of made was the implementation of an if statement to check if a question was received by the server and this if statement allows for answer to only be requested from the agent if a question was given. If no question was given the connection closes.

Possible Improvements

* The program could allow the agent more than one try at answering the security question right. This could be giving the agent another security question to answer and after a few tries closing the connection.