CSCI 3171 NETWORK COMPUTING
Assignment No. 3
Date Given: Sunday, October 24th, 2021
Due: Saturday, November 6th, 2021, 11.59 PM

Exercise 1: The objective of this exercise is to implement a Client-Server application that exchanges messages that are encrypted using Caesar cipher. It focuses on developing Client-Server applications using Java socket programming. Please read the lecture notes on Network Programming. Follow these steps:

a. Go to Brightspace -> Content -> Lecture Modules -> Module 3 -> Module 3-Part 5.
Download two files.

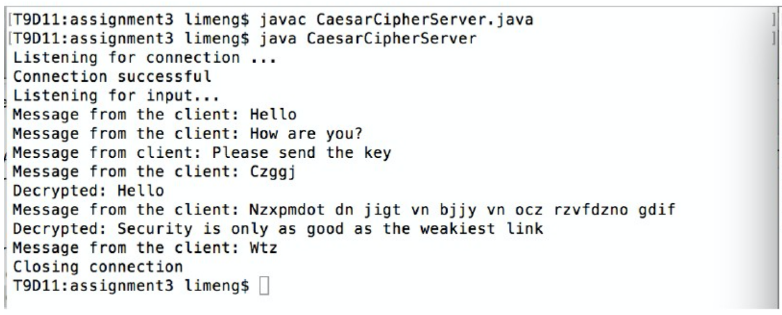
- EchoServer.java
- EchoClient.java

b. Caesar cipher is a simple alphabetic cipher that shifts the letters of the alphabet by k letters to the right where k is the "secret" key known only to the sender and the receiver. For example, if k = 3, the word "Java" will be encoded as "Mdyd". As another example, "ZEBRA" will be encrypted as "CHEUD" (notice that we wraparound to the beginning of the alphabet from Z).

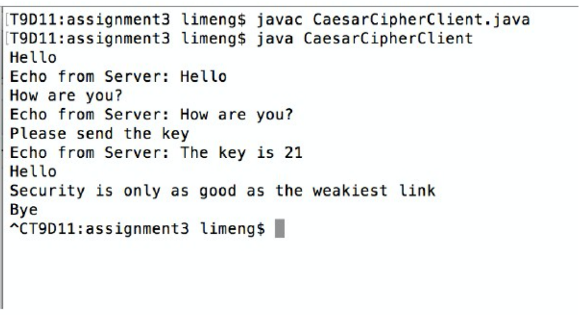
c. You are to write a client-server program in which the communication starts by the client asking the server for the key. A key (an integer between 1 and 25) is randomly generated by the server and sent to the client. (In practice, keys are transferred using a secure protocol, but for now to keep it simple we will just transfer the key in plaintext). Then every message that is sent from the client to the server must be encrypted using this key. Again, for the sake of simplicity, assume that only uppercase and lowercase letters are transmitted (no numbers or special characters). The message must be decrypted by the server and displayed. When the client enters "Bye", the connection terminates.

d. Provide three test cases for your program and take a screenshot of each test case. An example of the three test cases is shown below.

Test Case 1

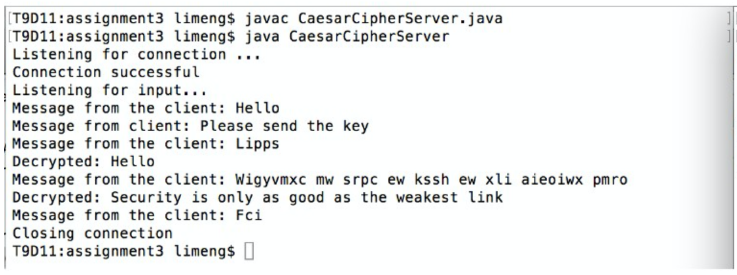


```
T9D11:assignment3 limeng$ javac CaesarCipherServer.java
T9D11:assignment3 limeng$ java CaesarCipherServer
Listening for connection ...
Connection successful
Listening for input...
Message from the client: Hello
Message from the client: How are you?
Message from client: Please send the key
Message from the client: Czggj
Decrypted: Hello
Message from the client: Nzxpmdot dn jigt vn bjjy vn ocz rzvfdzno gdif
Decrypted: Security is only as good as the weakest link
Message from the client: Wtz
Closing connection
T9D11:assignment3 limeng$
```

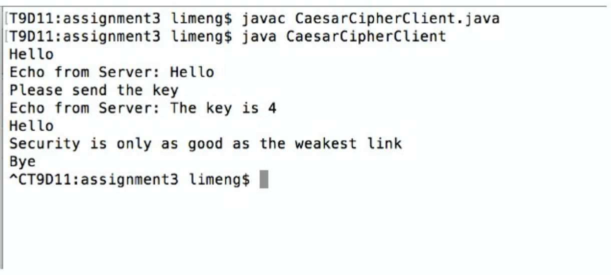


```
T9D11:assignment3 limeng$ javac CaesarCipherClient.java
T9D11:assignment3 limeng$ java CaesarCipherClient
Hello
Echo from Server: Hello
How are you?
Echo from Server: How are you?
Please send the key
Echo from Server: The key is 21
Hello
Security is only as good as the weakest link
Bye
^CT9D11:assignment3 limeng$
```

Test Case 2

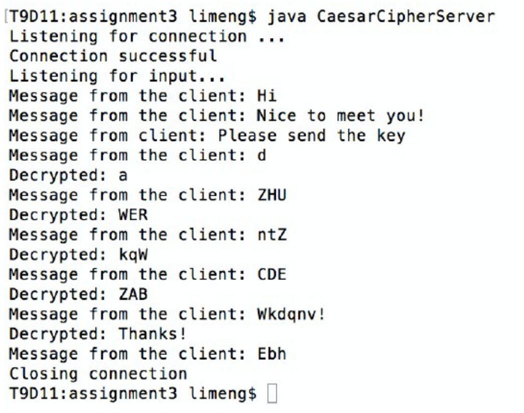


```
T9D11:assignment3 limeng$ javac CaesarCipherServer.java
T9D11:assignment3 limeng$ java CaesarCipherServer
Listening for connection ...
Connection successful
Listening for input...
Message from the client: Hello
Message from client: Please send the key
Message from the client: Lipps
Decrypted: Hello
Message from the client: Wigyvmxc mw srpc ew kssh ew xli aieoiwx pmro
Decrypted: Security is only as good as the weakest link
Message from the client: Fci
Closing connection
T9D11:assignment3 limeng$
```

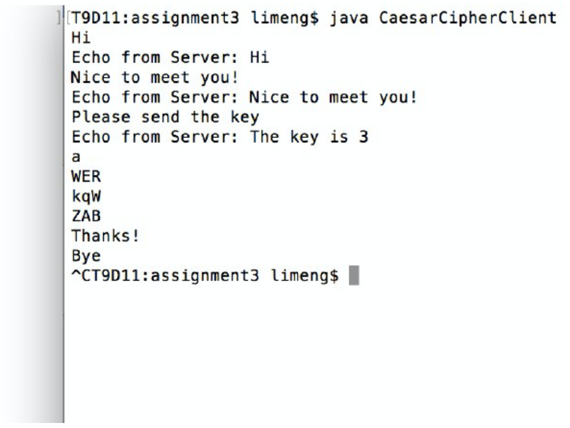


```
T9D11:assignment3 limeng$ javac CaesarCipherClient.java
T9D11:assignment3 limeng$ java CaesarCipherClient
Hello
Echo from Server: Hello
Please send the key
Echo from Server: The key is 4
Hello
Security is only as good as the weakest link
Bye
^CT9D11:assignment3 limeng$
```

Test Case 3



```
T9D11:assignment3 limeng$ java CaesarCipherServer
Listening for connection ...
Connection successful
Listening for input...
Message from the client: Hi
Message from the client: Nice to meet you!
Message from client: Please send the key
Message from the client: d
Decrypted: a
Message from the client: ZHU
Decrypted: WER
Message from the client: ntZ
Decrypted: kqW
Message from the client: CDE
Decrypted: ZAB
Message from the client: Wkdqnv!
Decrypted: Thanks!
Message from the client: Ebh
Closing connection
T9D11:assignment3 limeng$
```



```
T9D11:assignment3 limeng$ java CaesarCipherClient
Hi
Echo from Server: Hi
Nice to meet you!
Echo from Server: Nice to meet you!
Please send the key
Echo from Server: The key is 3
a
WER
kqW
ZAB
Thanks!
Bye
^CT9D11:assignment3 limeng$
```

Exercise 2: Now that you have seen how a simple encryption algorithm works, in this exercise you will conduct the wireshark analysis of a secure application layer protocol, SSH (Secure Shell).

a. Start Wireshark capture.
b. Connect to the timberlea.cs.dal.ca server using SSH from your terminal window. For this you would open a command window/Terminal, enter

   ssh -l <your CS username> timberlea.cs.dal.ca

   and enter your CS password. Then log out. Stop Wireshark capture.
c. Set the filter to ssh and provide a screen capture of the Wireshark interaction.
d. Identify and report some of the important SSH protocol messages.
e. Show the content of one encrypted packet.
f. Set the filter to tcp.port==<your client port number> and provide a screen captureof the Wireshark interaction.

g. Identify the TCP SYN, SYN-ACK and ACK segments and the FIN-ACK and ACK segments before the start of the SSH session and the end of the SSH session,respectively.

Note: The objective of this question is to get an overview of SSH. It is not necessary foryou at this point to understand the details of the encryption mechanisms and the key exchange algorithms used in SSH.

Submission:

Submit **one zip file** containing the following:

a) Commented CaesarCipherServer.java and CaesarCipherClient.java source codes for Exercise 1.

d) One word/text/pdf document containing the following:

i) Test cases with outputs of your program run for Exercise 1.

ii) Screenshot of the ssh dialog on the terminal window for Exercise 2.

iii) Screenshot of the Wireshark captures for Exercise 2.

iv) Answers to questions d, e and g in Exercise 2 (either marked on the screenshot or written separately)

Ensure that your Name and Banner ID appears as comments on top of each of the source code files. Also ensure that your Name and Banner ID appears on top of the word/text/pdf document.

Note: You must submit .java files so that the TAs are able to run your codes.

Late Submission Penalty: The assignment is due on Saturday at 11.59 PM. Late submissions up to 5 hours (4.59 AM on Sunday) will be accepted without penalty. After that, there will be a 10% late penalty per day on the mark obtained. For example, if you submit the assignment on Sunday at 12 noon and your score is 8/10, it will be reduced to 7.2/10. Submissions past five days after the grace submission time will not be accepted.