CSCI368 Network Security

Assignment 1 (20 Marks)

Submission Due: **4 FEB** 2024 23:55 (SG Time)

Objectives

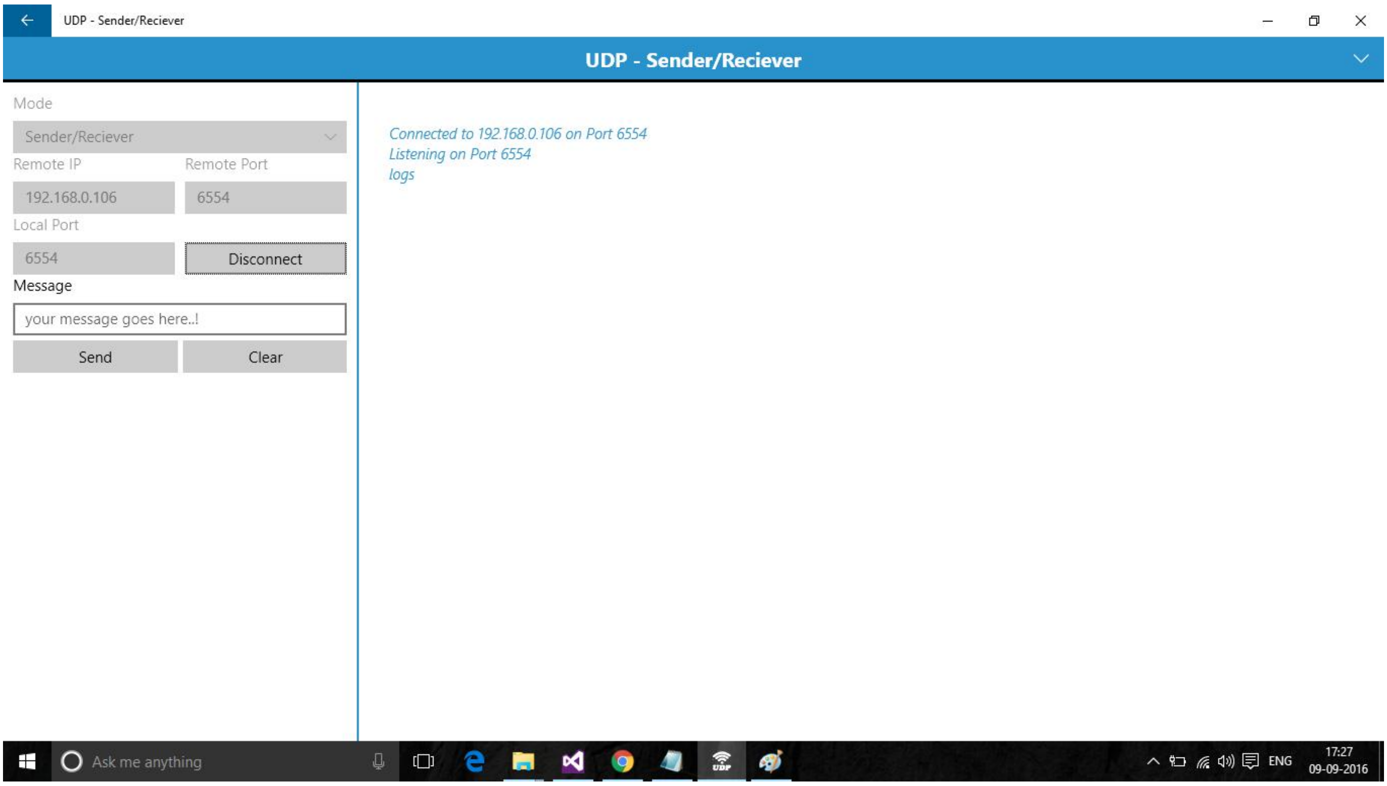
On completion of this assignment you should be able to:

- Understand some basic techniques for building a secure channel.
- Understand network programming.

Write (Java or C/C++ or others approved by the Tutor) UDP programs allowing two parties to establish a secure communication channel, which are executed by Alice and Bob, respectively.

Basics: (Reference Only)

References: https://apps.microsoft.com/store/detail/udp-senderreciever/9NBLGGH52BT0?hl=en-us&gl=US



UDP - Sender/Reciever

Mode: Sender/Reciever

Remote IP: 192.168.0.106

Remote Port: 6554

Local Port: 6554

Message: your message goes here..!

```
Connected to 192.168.0.106 on Port 6554
Listening on Port 6554
logs
```

The above is an app for communications between Alice and Bob using the UDP protocol. You should be family with this app and its function before doing this assignment.

This app, however, it is not secure. What you are going to do is to secure it. For simplicity, there is no GUI required in this assignment. That is, messages are simply typed on the sender's window and printed on the receiver's window. The looping should continue until the connection is terminated.

Idea:

When Alice(Bob) wants to communicate with Bob(Alice), she(he) needs to input:

- Remote IP, Remote Port, Remote **PK** **(receiver)**
- Local IP, Local Port, Local **PK** **(sender)**

The above info can be stored in a file and read it when using it. please use the local IP: 127.0.0.1 inside the file for simplifying the marking process.

Here, pk refers to the user's public key. That is, the secure communication requires that Alice and Bob know the other's public key first. Suppose that

- pk_R is the receiver's public key, and sk_R is the receiver's secret key.
- pk_S is the sender's public key and sk_S is the sender's secret key.

Adopted Cryptography includes:

- H, which is a cryptography hash function (the SHA-1 hash function).
- E and D, which are encryption algorithm and decryption algorithm of symmetric-key encryption (AES for example)
- About the key pair, sk=x and pk=g^x. (based on cyclic groups)

You can use an open-source crypto library or some open-source code to implement the above cryptography.
What you need to code is the following algorithms.

When the sender inputs a message M and clicks "Send", the app will do as follows before sending it to the receiver.

- Choose a random number r (nonce) from Z_p and compute g^r and TK=(pk_R)^r.
- Use TK to encrypt M denoted by C=E(TK, M)
- Compute LK=(pk_R)^{sk_s}.
- Compute MAC=H(LK || g^r || C || LK). Here, || denotes the string concatenation.
- Send (g^r, C, MAC) to the receiver.
- The sender part should display M and (g^r, C, MAC)

That is, for security purpose, M is replaced with (g^r, C, MAC)

Note: TK can be seen as a shared secret key that can be computed by both two parties.

When the receiver receives (g^r, C, MAC) from the sender, the app will do as follows.

- Compute TK=(g^r)^{sk_R}.
- Compute LK=(pk_S)^{sk_R}
- Compute MAC'=H(LK || g^r || C || LK). Here, || denotes the string concatenation.
- If MAC=MAC', go to next step. Otherwise, output "ERROR"
- Compute M'=D(TK, C) .

The receiver part should display

The decryption on
(g^r, C, MAC)
is
M' (or ERROR)

Note: the receiver can reply the message. The receiver becomes the sender, and the seconder becomes receiver.

Coding requirement:

You can use any open-source code as you like. You can use a crypto library or some open-source code to implement the encryption and hashing functions and the related group generation and key pair generation. **You should cite the source if you use a downloaded code.**

Files to be submitted:

All source codes.

A readme file (text/ACSII only): instructions about how to compile and run your code.

Submission

Compress all the files to be submitted into a zip file and submit it via the submission link provided in the Moodle site.

Late Submission: Penalty is 25% deduction per day (including weekends) unless Academic Consideration is granted.

Marking

Mark distribution:

1. UDP connection: 2 marks
2. The info sent from the sender is correctly generated: 9 marks
3. Data can be successfully decrypted and displayed: 9 marks

Plagiarism

A plagiarised assignment will receive a zero mark and be penalised according to the university rules. Plagiarism detection software may be used.