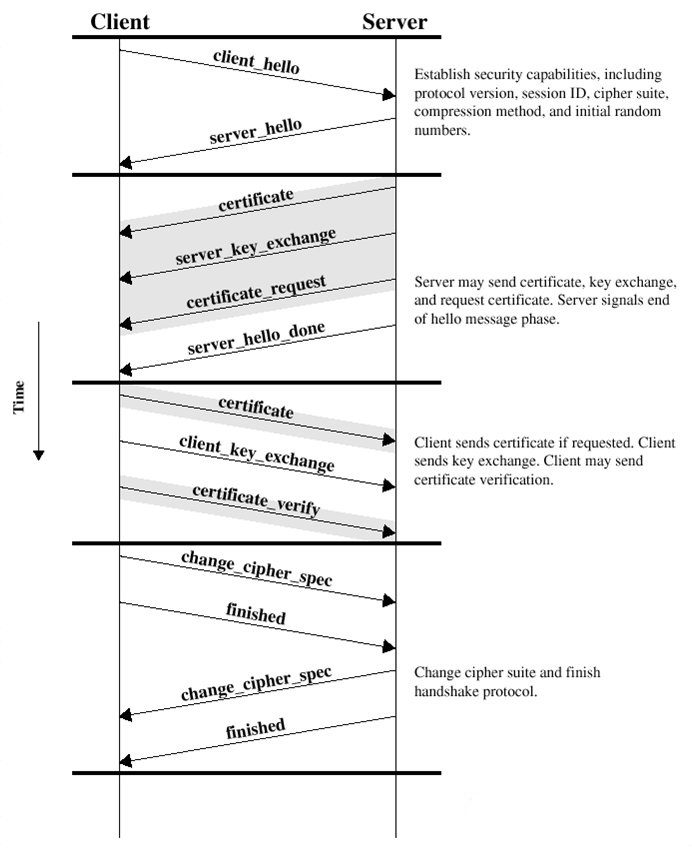
**TLS Project**

**The Goals of the Projec**t

A simple simulative realization of the TLS process and the newest TLS 1.3 is prefered. Through this project, you are expected to have a good understanding of the TLS. If you are interested, you can have a further study of the TLS Protocols Draft RFC8446. (**Reference)**



**What the Programs should Realize**

The Project should include two parts: Client and Server

.

1. The Client and the Server should set up a connection to communicate with each other. You can realize this by sockets.
2. The Client and the Server should agree on the session key for encryption, the session key should displayed on the screen or user interface of the server. Note: the session key is mainly determined by Client.

One simple method is as follows:

First, the Client produces the session key, then using the Server’s public key, the client encrypts the session key and sends it to the Server

You are appreciated to think out better ways to the session key production.

1. The client encrypts one string, computes the MAC and sends them the server
2. The Server decrypts, verifies the MAC.

**Recommendation**

1. the string encryption methods should be symmetric such as DES,3DES, which is much faster than asymmetric .
2. you can use SHA-256 to compute the MAC.

**What you Should hand in**

1.A detailed document explaining your design, meanwhile how to produce the session key must be included.

2. The source code

3. The executable program and a manual to it

**High-quality programming**

Pay attention to the quality of your code. You can make a brief introduction and discussion about Heartbleed (<https://heartbleed.com/>) in your report.

**This project will start from Oct.26, and last until Nov.16.**

You must compress all your files(codes, design document, executable program) into a .zip file, and the filename is the combination of your student number and name. For example, “192053017-YourName.zip”.

**Programs tools**

You can use some libraries of TLS which depends on the programing language used to implement. These libraries include many algorithms such DES, MD5 and so on .You should use them without writing owe-level cods.

**Warning: DO IT YOURSELF.**

**What your score composed of**

Correctness of the executable program: 60%

Client and Server connection 15%

Session key production 15%

Encryprion and Decryption 15%

MAC and verify 15%

Readability of the document and source code: 40%

Design document: 15%

Manual: 10%

Source code: 15%

The Client and Server may communicate each other to agree on the algorithms for encryption, MAC and so on before the session key production. The agreed algorithms should be displayed on the screen or user interface of the client

**Reference:**

You can refer to RFC 8446 <https://datatracker.ietf.org/doc/rfc8446/>