ENPM693 – Network Security

Section: 0101

Homework – 6

Name: Syed Mohammad Ibrahim

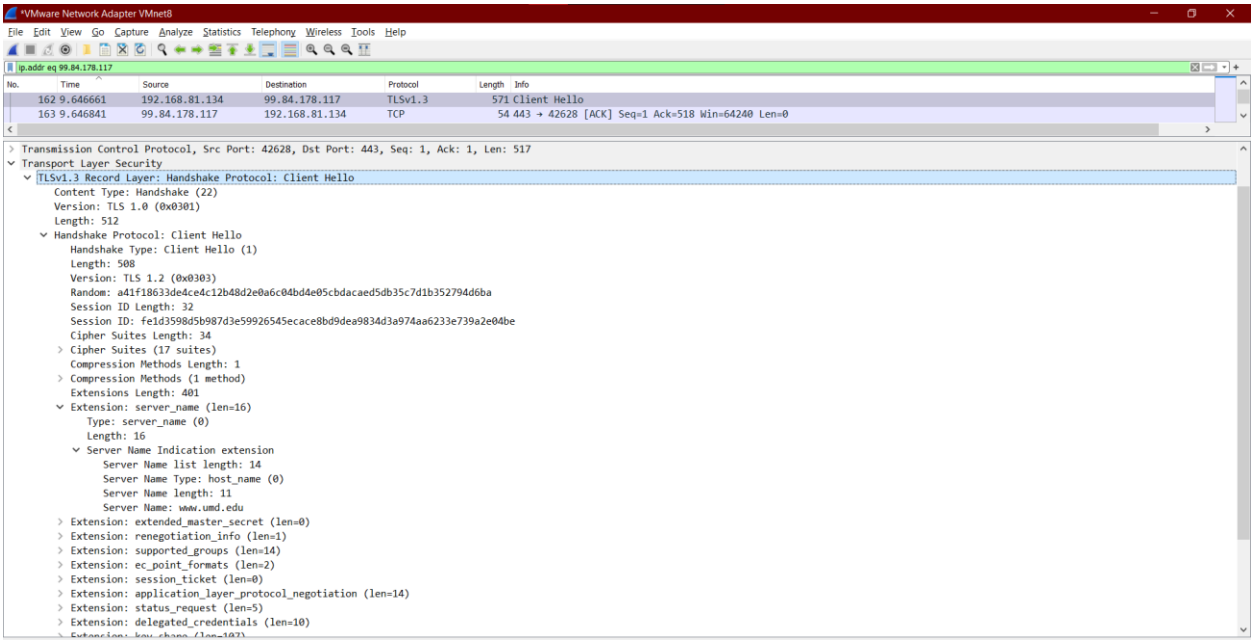
UID: *iamibi*

UID Number: 118428369

Summary

This document outlines the summary of TLS message exchanges between my system and the website https://www.umd.edu:

Client Request to Server



```
▾ Extension: key_share (len=107)
    Type: key_share (51)
    Length: 107
  ▾ Key Share extension
      Client Key Share Length: 105
    › Key Share Entry: Group: x25519, Key Exchange length: 32
    › Key Share Entry: Group: secp256r1, Key Exchange length: 65
▾ Extension: supported_versions (len=5)
    Type: supported_versions (43)
    Length: 5
    Supported Versions length: 4
    Supported Version: TLS 1.3 (0x0304)
    Supported Version: TLS 1.2 (0x0303)
› Extension: signature_algorithms (len=24)
▾ Extension: psk_key_exchange_modes (len=2)
    Type: psk_key_exchange_modes (45)
    Length: 2
    PSK Key Exchange Modes Length: 1
    PSK Key Exchange Mode: PSK with (EC)DHE key establishment (psk_dhe_ke) (1)
```

The client initiates a request with a “Client Hello” message with the server and adds additional information, few of which are:

Protocol Version

The TLS Protocol version used here is **TLSv1.3**

Session ID

Session identifier is an arbitrary byte sequence generated by client to maintain connection with the server.

Cipher Suites

This informs the server as to what acceptable cipher suites are available which can be used by server to determine the security that needs to be established. In my case, the client sends a list of 17 supported cipher suites to the server.

Client Random

This is a byte sequence that is generated by client for every connection. In TLSv1.3, the random also contains a timestamp value attached to it in a hashed form.

Server Name

The client sends the information about which server it is trying to access as part of *server_name* object. In my case, it was www.umd.edu

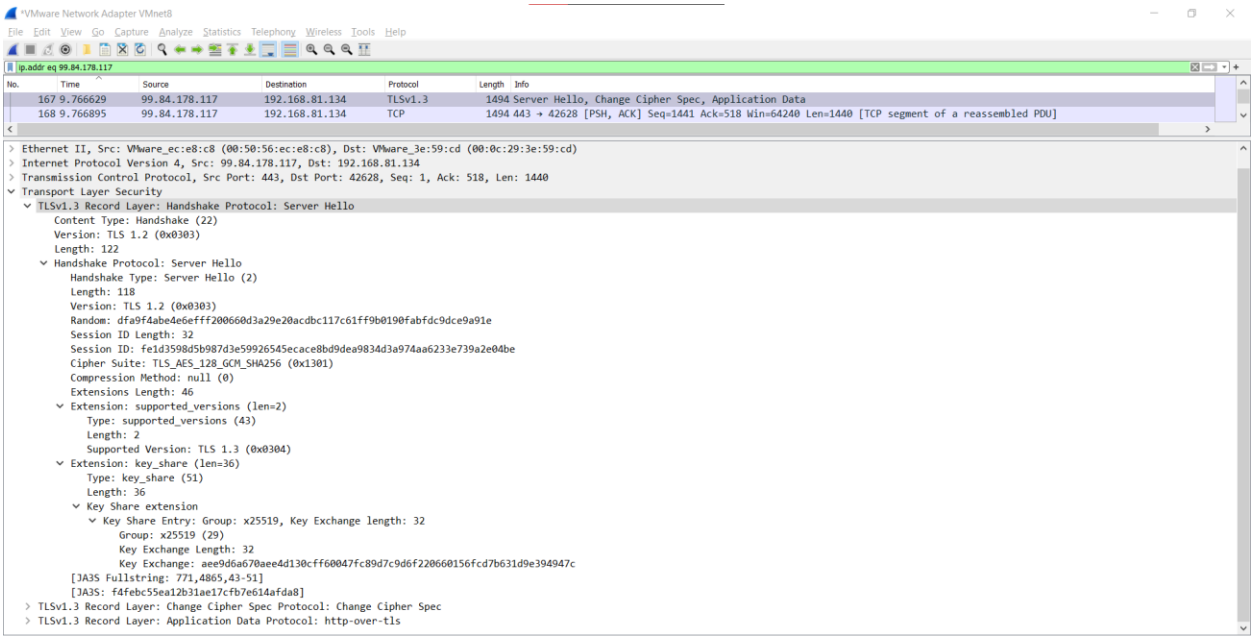
Key Exchange

The client attaches the information as to which mode will it be using to make the Key Exchange for secure communication if cipher keys. In my case, it states PSK with ECDHE.

Supported Versions

The client sends out the information about the compatibility of the TLS versions and which versions are supported by it so that the server can identify an appropriate algorithm to confirm on.

Server Response to Client



The server return's a "Server Hello" in response to the client's request. Few of the fields that are returned as part of the response are:

Cipher Suite

The server checks the client's Cipher Suite and then decides on which one to use from the given list. In my case, the server chose TLS_AES_128_GCM_SHA256 scheme.

Supported Version

The server finalizes the TLS version it agrees on with the client and set that as part of the response. In my case, it is TLSv1.3.

Server Key Share

Server shares its encryption key with the client using a key exchange protocol so that the client can decrypt it.

Session Identifier

Server returns the same Session ID as part of the response message to client that the client originally sent it in the request to maintain a session.

Server Random

This is a byte sequence that is generated by server for every connection. In TLSv1.3, the random also contains a timestamp value attached to it in a hashed form.

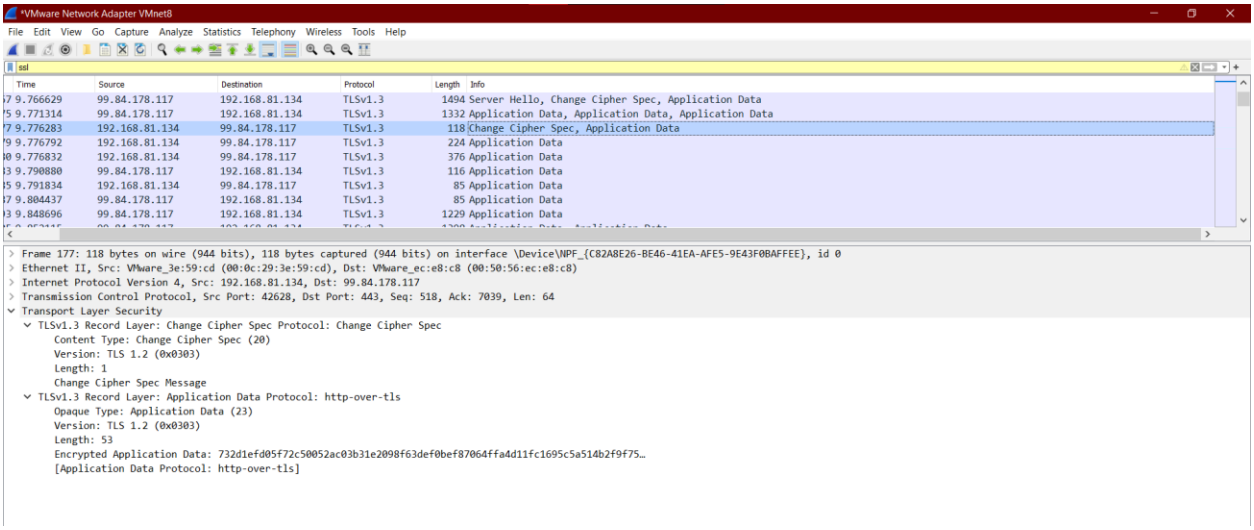
Change Cipher Spec

The server adds the finalized cipher spec as part of the response so that client can use it for the next messages.

Encrypted Data

The server starts sending encrypted data from here on as part of the response along with a “Server Finished” message.

Client Response to Server



The client now verifies the Certificate shared by the server and generates the Symmetric Key from the Key Share provided by the server and responds with the following:

Change Cipher Spec

Client accepted Cipher Spec which is now agreed by both client and server.

Client Finished

Client responds with a “Client Finished” message along with encrypted data as part of the response. It indicates that client is done with setting up communication and it starts sharing the data as encrypted messages.