# For each of the first 8 Ethernet frames, specify the source of the frame (client or server), determine the number of SSL records that are included in the frame, and list the SSL record types that are included in the frame. Draw a timing diagram between client and server, with one arrow for each SSL record.

图片包含 表格

描述已自动生成

As the above figure shows, from the source 128.238.38.162 to the destination 216.75.194.220 is the client to the sever. The first 8 Ethernet frames include Client Hello, Server Hello, Certificate, Server Hello Done, Client Key Exchange, Change Cipher Spec, Encrypted Handshake Message, Change Cipher Spec, Encrypted Handshake Message, Application Data \*3. The number of the SSL records is 12.

图示

中度可信度描述已自动生成

# Each of the SSL records begins with the same three fields (with possibly different values). One of these fields is “content type” and has length of one byte. List all three fields and their lengths.

## Client Hello

图形用户界面, 应用程序, Word

描述已自动生成

Length: 0x804c (2 bytes) = 76

Type: 0x01 (1 bytes) = 1 (Client Hello)

Version: 0x 0300 (2 bytes) = SSL 3.0

All of them are counted 5 bytes.

## Server Hello

图形用户界面, 应用程序

描述已自动生成

Length: 0x004a (2 bytes) = 74

Type: 0x16 (1 bytes) = 22 (Handshake)

Version: 0x 0300 (2 bytes) = SSL 3.0

## Certificate, Server Hello Done

图形用户界面

描述已自动生成

### Certificate

Length: 0x0a83 (2 bytes) = 2691

Type: 0x16 (1 bytes) = 22 (Handshake)

Version: 0x 0300 (2 bytes) = SSL 3.0

### Server Hello Done

Length: 0x0004 (2 bytes) = 4

Type: 0x16 (1 bytes) = 22 (Handshake)

Version: 0x 0300 (2 bytes) = SSL 3.0

## Client Key Exchange, Change Cipher Spec, Encrypted Handshake Message

图形用户界面, 应用程序

描述已自动生成

### Client Key Exchange

Length: 0x0084 (2 bytes) = 132

Type: 0x16 (1 bytes) = 22 (Handshake)

Version: 0x 0300 (2 bytes) = SSL 3.0

### Change Cipher Spec

Length: 0x0001 (2 bytes) = 1

Type: 0x14 (1 bytes) = 20 (Change Cipher Spec)

Version: 0x 0300 (2 bytes) = SSL 3.0

### Encrypted Handshake Message

Length: 0x0038 (2 bytes) = 56

Type: 0x16 (1 bytes) = 22 (Handshake)

Version: 0x 0300 (2 bytes) = SSL 3.0

## Change Cipher Spec, Encrypted Handshake Message

图形用户界面, 应用程序

描述已自动生成

### Change Cipher Spec

Length: 0x0001 (2 bytes) = 1

Type: 0x14 (1 bytes) = 20 (Change Cipher Spec)

Version: 0x 0300 (2 bytes) = SSL 3.0

### Encrypted Handshake Message

Length: 0x0038 (2 bytes) = 56

Type: 0x16 (1 bytes) = 22 (Handshake)

Version: 0x 0300 (2 bytes) = SSL 3.0

## Application Data

图形用户界面, 文本, 应用程序

描述已自动生成

Length: 0x02eb (2 bytes) = 747

Type: 0x17 (1 bytes) = 23 (Application Data)

Version: 0x 0300 (2 bytes) = SSL 3.0

## Application Data

图形用户界面, 文本, 应用程序

描述已自动生成

Length: 0x00d5 (2 bytes) = 213

Type: 0x17 (1 bytes) = 23 (Application Data)

Version: 0x 0300 (2 bytes) = SSL 3.0

## Application Data

图形用户界面, 文本, 应用程序

描述已自动生成

Length: 0x2010 (2 bytes) = 8208

Type: 0x17 (1 bytes) = 23 (Application Data)

Version: 0x 0300 (2 bytes) = SSL 3.0

# Expand the ClientHello record. (If your trace contains multiple ClientHello records, expand the frame that contains the first one.) What is the value of the content type?

日历

描述已自动生成

# Does the ClientHello record contain a nonce (also known as a “challenge/random”)? If so, what is the value of the challenge in hexadecimal notation?

表格

中度可信度描述已自动生成

The Client Hello has the nonce. The challenge is 0x66 df 78 4c 04 8c d6 04 35 dc 44 89 89 46 99 09.

# Does the ClientHello record advertise the cyber suites it supports? If so, in the first listed suite, what are the public-key algorithm, the symmetric-key algorithm, and the hash algorithm?

手机屏幕截图

描述已自动生成

Yes, the first cyber suites is TLS\_RSA\_WITH\_RC4\_128\_MD5.

public-key algorithm: RSA

symmetric-key algorithm: RC4 with a 128-bit key

hash algorithm: MD5

# Locate the ServerHello SSL record. Does this record specify a chosen cipher suite? What are the algorithms in the chosen cipher suite?

图形用户界面, 文本, 应用程序, 电子邮件

描述已自动生成

Yes, it choose the TLS\_RSA\_WITH\_RC4\_128\_MD5 as the cipher suite.

# Does this record include a nonce? If so, how long is it? What is the purpose of the client and server nonces in SSL?

Yes, it has 32 bytes, which includes GMT Unix Time and Random Bytes. The Random Bytes is 28 Bytes. The purpose of the of the client and server nonces in SSL are to prevent the secret of the client be stolen. And it can prevent the Replay attack.

图片包含 日历

描述已自动生成

# Does this record include a session ID? What is the purpose of the session ID?

图形用户界面, 文本, 应用程序

描述已自动生成

Yes, it records a session ID, which is to tag this process and ensure the server to recognize the specific client.

# Does this record contain a certificate, or is the certificate included in a separate record. Does the certificate fit into a single Ethernet frame?

No, the certificate included in a separate record named Certificate, which do not also fit into a single Ethernet frame, the frame contains Certificate and Server Hello Done.

# Locate the client key exchange record. Does this record contain a pre-master secret? What is this secret used for? Is the secret encrypted? If so, how? How long is the encrypted secret?

表格

描述已自动生成

It has the Encrypted Premaster, which is used for transmitting the premaster secret to server. When the client generates a random number, it uses the public key to encrypt the random number as encrypted premaster secret and transmit it to the server. And then the server and client use the private key to decrypt the encrypted premaster secret as premaster secret, which is used for encrypting the communications. The length of the encrypted secret is 128 Bytes.

1.How is the application data being encrypted

The application data is encrypted by session key, which is symmetric encryption.

Master Key = first nonce(from client) + sencond nonce(from serve)+ premaster key(from client and encrypt by public key)

Then use the master key to generate a session key to ensure communication security.

2. the encryption algorithm used in my trace

Cipher Suite: TLS\_ECDHE\_RSA\_WITH\_AES\_256\_GCM\_SHA384 (0xc030)

protocol:TLS

Public-key algorithm:RSA

Symmetric-key algorithm:AES\_256\_GCM

Hash algorithm: SHA384

3.Do the records containing application data include a MAC?

Yes, the records contain MAC.

4.Does Wireshark distinguish between the encrypted application data and the MAC?

No.

# What is the purpose of the Change Cipher Spec record? How many bytes is the record in your trace?

表格

低可信度描述已自动生成

The purpose of the Change Cipher Spec is to use the premaster secret and communicate with the client and server, which can improve the security of the communications. The Change Cipher Spec has 6 Bytes.

# In the encrypted handshake record, what is being encrypted? How?

图片包含 表格

描述已自动生成

The encrypted handshake record contains the Pre-Master Secret, Key Exchange Parameters, and the Server Certificate, all of which are encrypted through Hash and then became the encrypted handshake record. This record is to make sure the correct of the handshake message.

# Does the server also send a change cipher record and an encrypted handshake record to the client? How are those records different from those sent by the client?

日历

描述已自动生成

Yes, the server also sends the change cipher record and encrypted handshake record to the client. The difference between the client and the server is that the encrypted handshake record in client contain the encrypted nonce, Key Exchange Parameters, and Server Certificate, while the encrypted handshake record in server contains the Premaster secret and certificate without the nonce.

# How is the application data being encrypted? Do the records containing application data include a MAC? Does Wireshark distinguish between the encrypted application data and the MAC?

图形用户界面, 文本, 应用程序, 电子邮件

描述已自动生成

After the handshake step, the server and the client would generate the master secret through the premaster secret, using it each of which can generate the public key, private key and the MAC key. The Mac Key will generate the MAC through hash the message. And the MAC will follow the encrypted message to be send to the destination. But the Wireshark cannot distinguish between the encrypted application data and the MAC.

# Comment on and explain anything else that you found interesting in the trace.

1.How is the application data being encrypted

The application data is encrypted by session key, which is symmetric encryption.

Master Key = first nonce(from client) + sencond nonce(from serve)+ premaster key(from client and encrypt by public key)

Then use the master key to generate a session key to ensure communication security.

2. the encryption algorithm used in my trace

Cipher Suite: TLS\_ECDHE\_RSA\_WITH\_AES\_256\_GCM\_SHA384 (0xc030)

protocol:TLS

Public-key algorithm:RSA

Symmetric-key algorithm:AES\_256\_GCM

Hash algorithm: SHA384

3.Do the records containing application data include a MAC?

Yes, the records contain MAC.

4.Does Wireshark distinguish between the encrypted application data and the MAC?

No.

Question 3

4 Points

Expand the ClientHello record. (If your trace contains multiple ClientHello records, expand the frame that contains the first one.) In your trace, specify the first 5 bytes in this record. What do they represent?

16 03 01 01 65

16 represents the content type: Handshake (22).

03 01 represents the version: TLS 1.2 (0x0303)

01 65 represents the length: 353