山东大学 计算机 学院

计算机网络 课程实验报告

学号：	姓名：	班级：
实验题目： 实验十二 SSL		
实验学时：2h	实验日期：2023.05.22	
实验目的： 学习有关 SSL 的相关知识		
硬件环境： Windows10 家庭版		
软件环境： Wireshark		

实验步骤与内容：

实验内容：

1. 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.

2. 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.

3. 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?

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

5. 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?

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

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

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

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

10. 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?

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

12. In the encrypted handshake record, what is being encrypted? How?

13. 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?

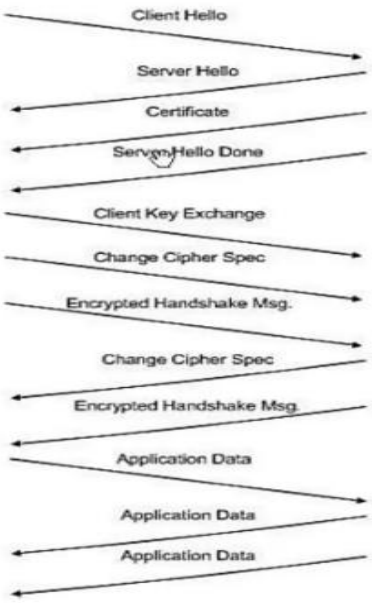
14. How is the application data being encrypted? Do the records containing application data include a MAC? Does Wireshark distinguish between the encrypted application data and the MAC?

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

实验步骤：

1.

Frame # in Ethereal	Source	# of SSL Records	List of SSL Records
106	Client	1	Client Hello
108	Server	1	Server Hello
111	Server	2	Certificate Server Hello Done
112	Client	3	Client Key Exchange Change Cipher spec Encrypted Handshake message
113	Server	2	Change Cipher spec Encrypted Handshake message
114	Client	1	Application Data
122	Server	1	Application Data
149	Server	1	Application Data



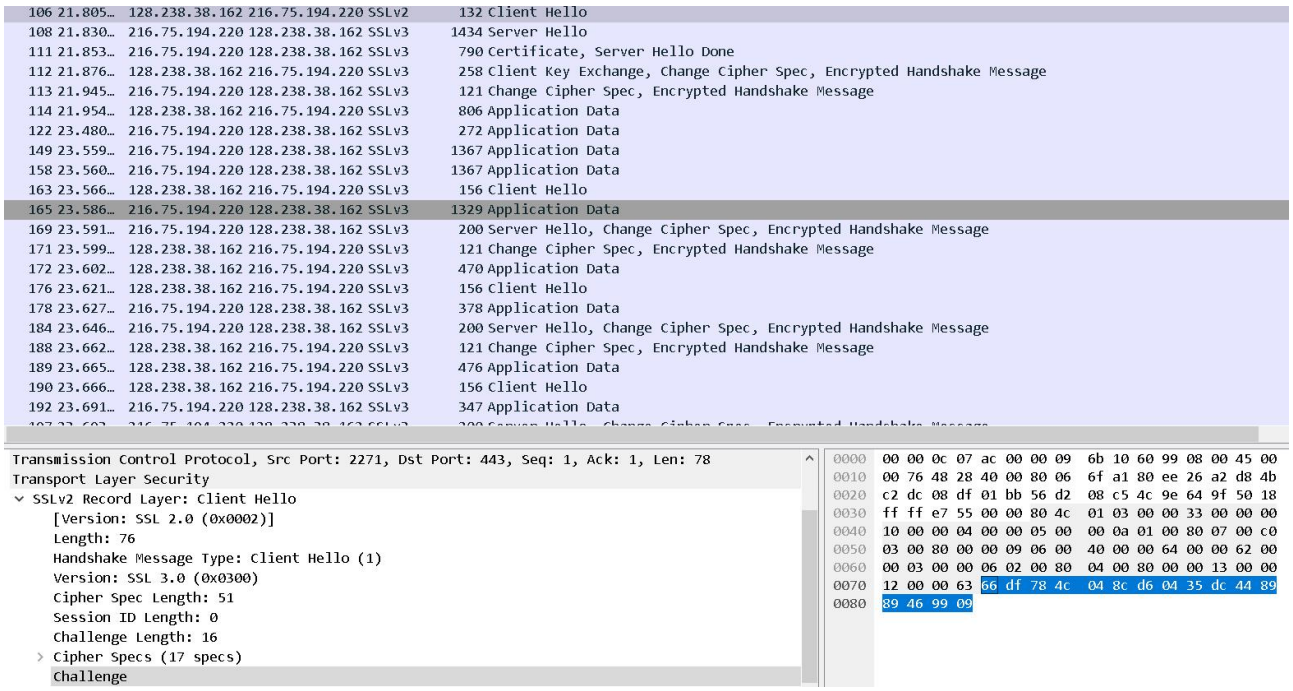
2. 三个字段分别是 Content Type：1byte，Version ：2bytes，Length ：2bytes。

3. 内容类型的值是 22。

```
  165 23.586…  216.75.194.220 128.238.38.162 SSLv3     1329 Application Data
  169 23.591…  216.75.194.220 128.238.38.162 SSLv3      200 Server Hello, Change Cipher Spec, Encrypted H
  171 23.599…  128.238.38.162 216.75.194.220 SSLv3      121 Change Cipher Spec, Encrypted Handshake Messa
  172 23.602…  128.238.38.162 216.75.194.220 SSLv3      470 Application Data
  176 23.621…  128.238.38.162 216.75.194.220 SSLv3      156 Client Hello
  178 23.627…  216.75.194.220 128.238.38.162 SSLv3      378 Application Data
  184 23.646…  216.75.194.220 128.238.38.162 SSLv3      200 Server Hello, Change Cipher Spec, Encrypted H
  188 23.662…  128.238.38.162 216.75.194.220 SSLv3      121 Change Cipher Spec, Encrypted Handshake Messa
  189 23.665…  128.238.38.162 216.75.194.220 SSLv3      476 Application Data
  190 23.666…  128.238.38.162 216.75.194.220 SSLv3      156 Client Hello
  192 23.691…  216.75.194.220 128.238.38.162 SSLv3      347 Application Data
```

```
> Frame 163: 156 bytes on wire (1248 bits), 156 bytes captured (1248 bits)
> Ethernet II, Src: IBM_10:60:99 (00:09:6b:10:60:99), Dst: All-HSRP-routers_00 (00:00:0c:07:ac:00)
> Internet Protocol Version 4, Src: 128.238.38.162, Dst: 216.75.194.220
> Transmission Control Protocol, Src Port: 2272, Dst Port: 443, Seq: 1, Ack: 1, Len: 102
v Transport Layer Security
  v SSLv3 Record Layer: Handshake Protocol: Client Hello
      Content Type: Handshake (22)
      Version: SSL 3.0 (0x0300)
      Length: 97
    > Handshake Protocol: Client Hello
```

4. 是的，它在十六进制中的值是 0x66df784c048cd60435dc448989469909。



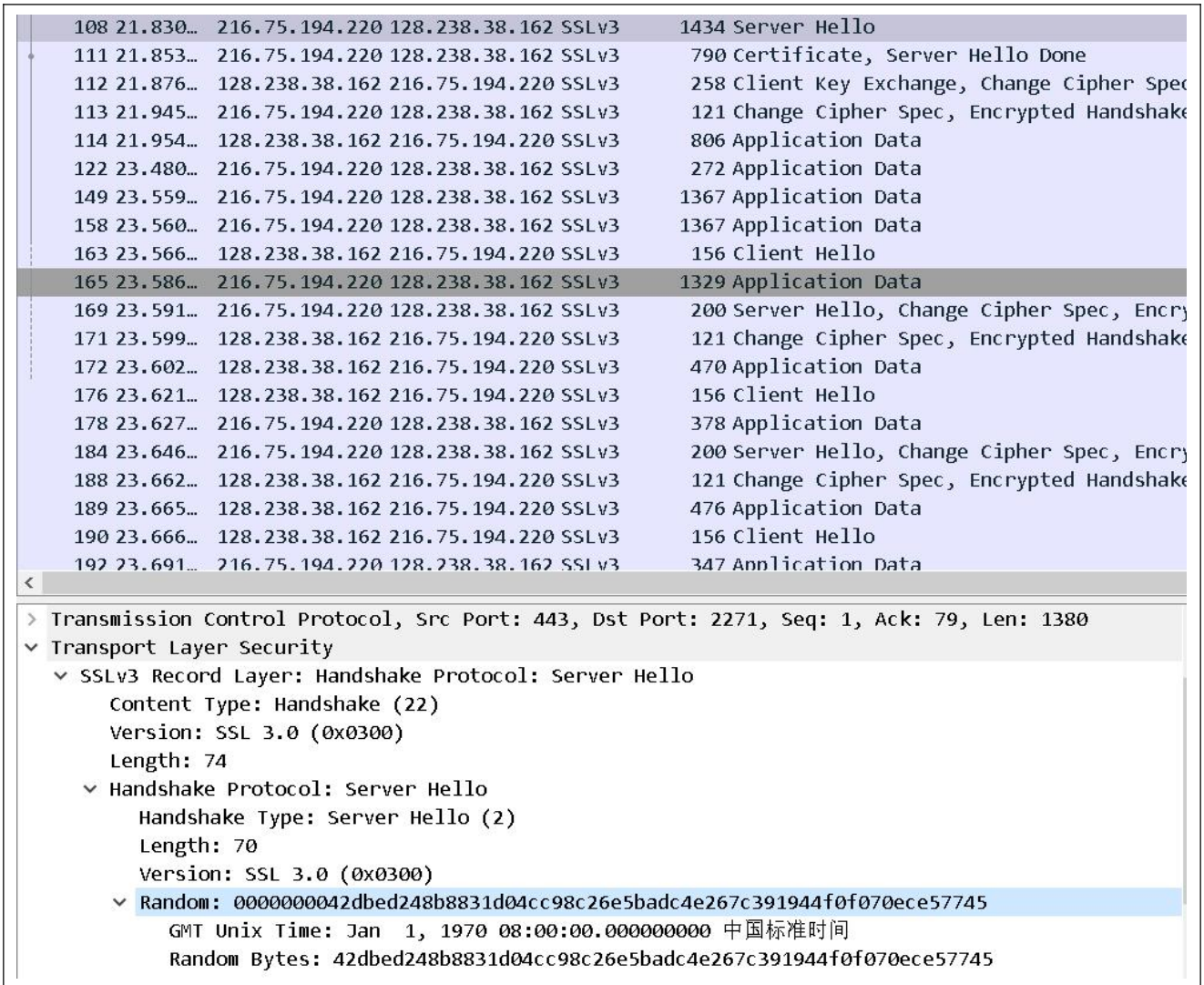
5. 哈希算法是 MD5，公钥算法是 RSA，对称密钥算法是 RC4_128。

```
Challenge Length: 16
Cipher Specs (17 specs)
    Cipher Spec: TLS_RSA_WITH_RC4_128_MD5 (0x000004)
    Cipher Spec: TLS_RSA_WITH_RC4_128_SHA (0x000005)
    Cipher Spec: TLS_RSA_WITH_3DES_EDE_CBC_SHA (0x00000a)
    Cipher Spec: SSL2_RC4_128_WITH_MD5 (0x010080)
    Cipher Spec: SSL2_DES_192_EDE3_CBC_WITH_MD5 (0x0700c0)
    Cipher Spec: SSL2_RC2_128_CBC_WITH_MD5 (0x030080)
    Cipher Spec: TLS_RSA_WITH_DES_CBC_SHA (0x000009)
    Cipher Spec: SSL2_DES_64_CBC_WITH_MD5 (0x060040)
    Cipher Spec: TLS_RSA_EXPORT1024_WITH_RC4_56_SHA (0x000064)
    Cipher Spec: TLS_RSA_EXPORT1024_WITH_DES_CBC_SHA (0x000062)
    Cipher Spec: TLS_RSA_EXPORT_WITH_RC4_40_MD5 (0x000003)
```

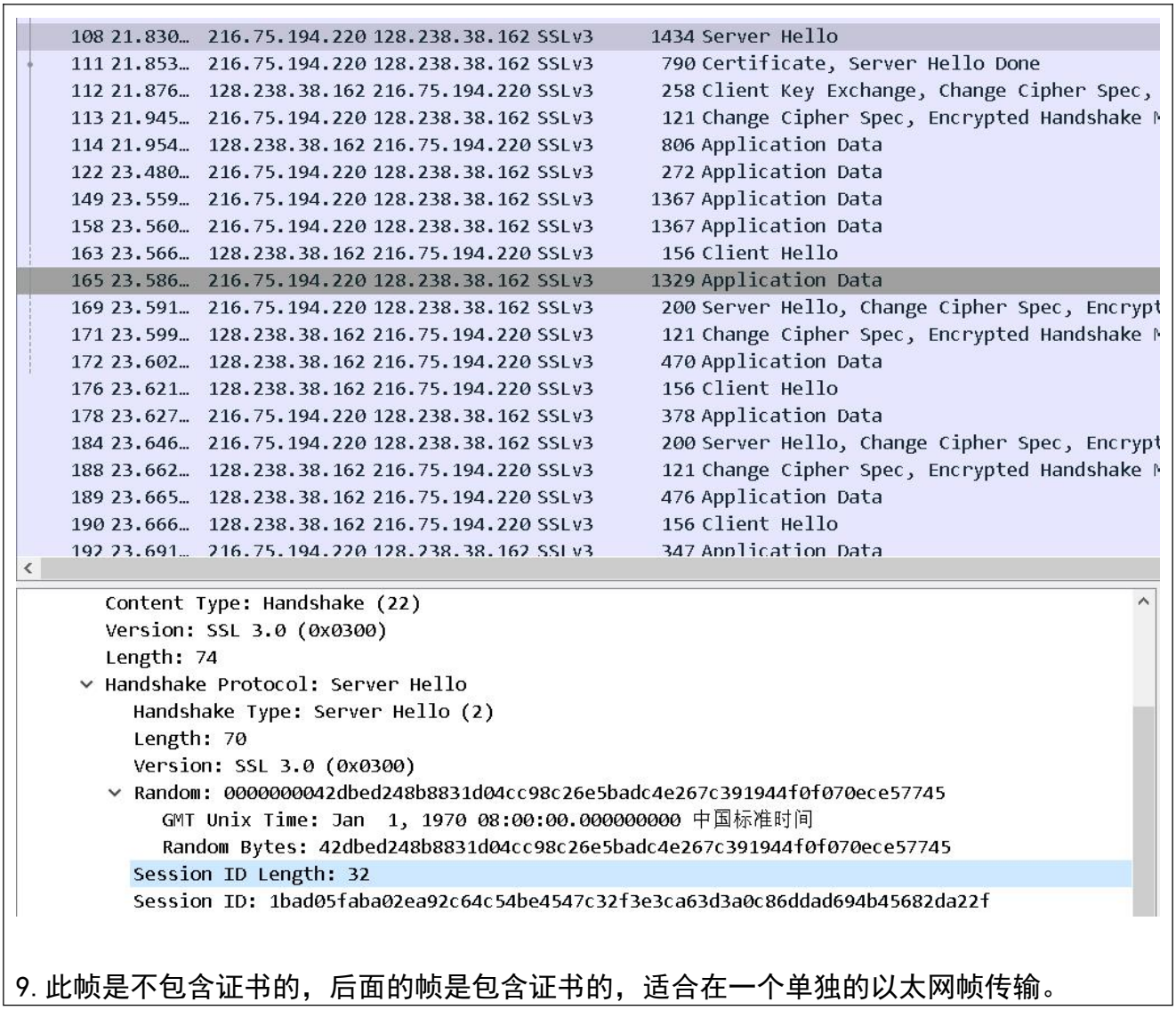
6. 指定了加密算法，其中公钥算法是 RSA, 对称密钥算法是 RC4_128, 哈希算法是 MD5。

```
Content Type: Handshake (22)
Version: SSL 3.0 (0x0300)
Length: 74
Handshake Protocol: Server Hello
    Handshake Type: Server Hello (2)
    Length: 70
    Version: SSL 3.0 (0x0300)
    Random: 0000000042dbed248b8831d04cc98c26e5badc4e267c391944f0f070ece57745
    Session ID Length: 32
    Session ID: 1bad05faba02ea92c64c54be4547c32f3e3ca63d3a0c86ddad694b45682da22f
    Cipher Suite: TLS_RSA_WITH_RC4_128_MD5 (0x0004)
    Compression Method: null (0)
    [JA3S Fullstring: 768,4,]
    [JA3S: 1f8f5a3d2fd435e36084db890693eafd]
```

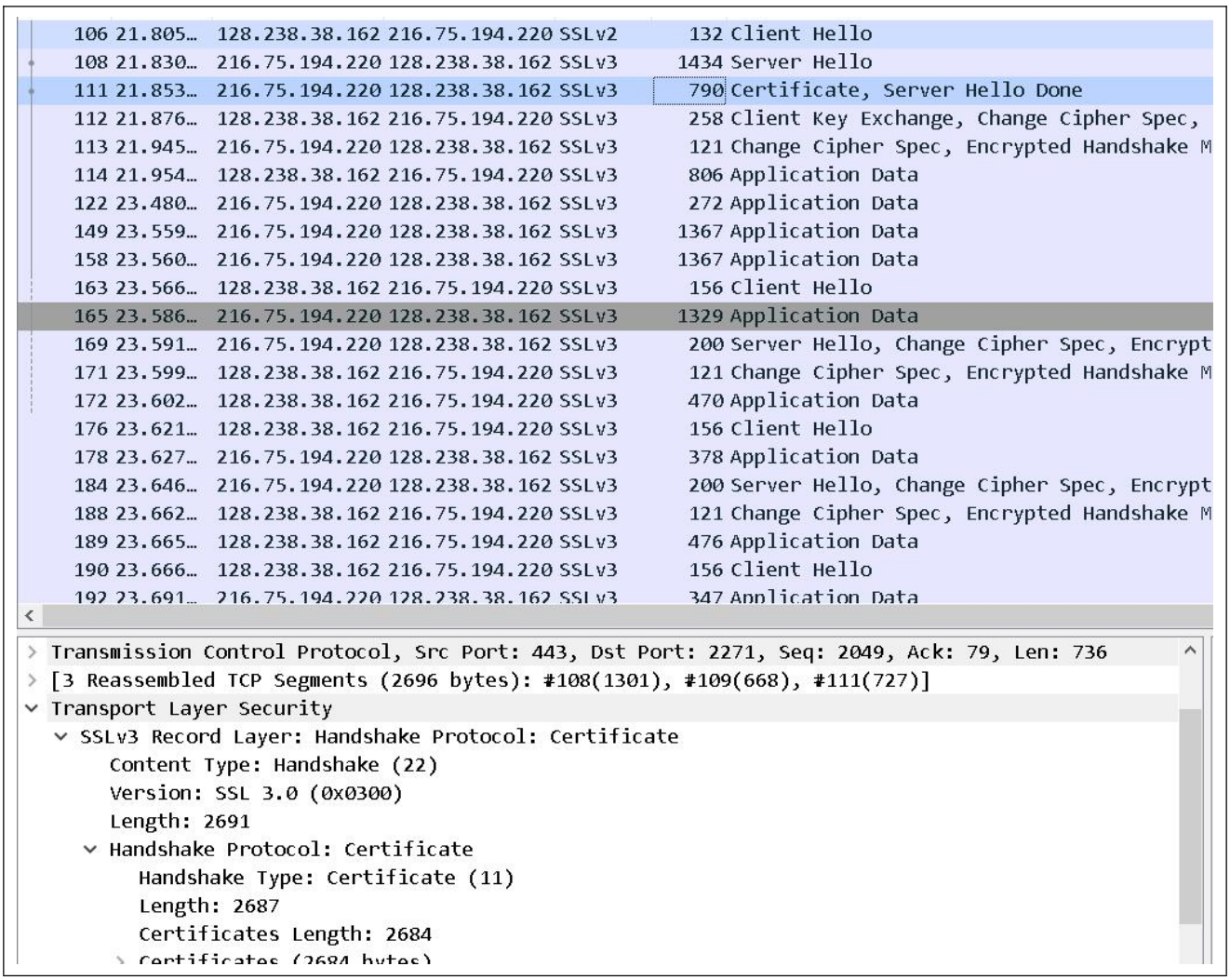
7. 包括了随机数，其中随机数长度为 32 字节，28 个字节是数据，4 个字节是时间，目的是多次随机数生成为未来生成对话密钥提高安全性能。

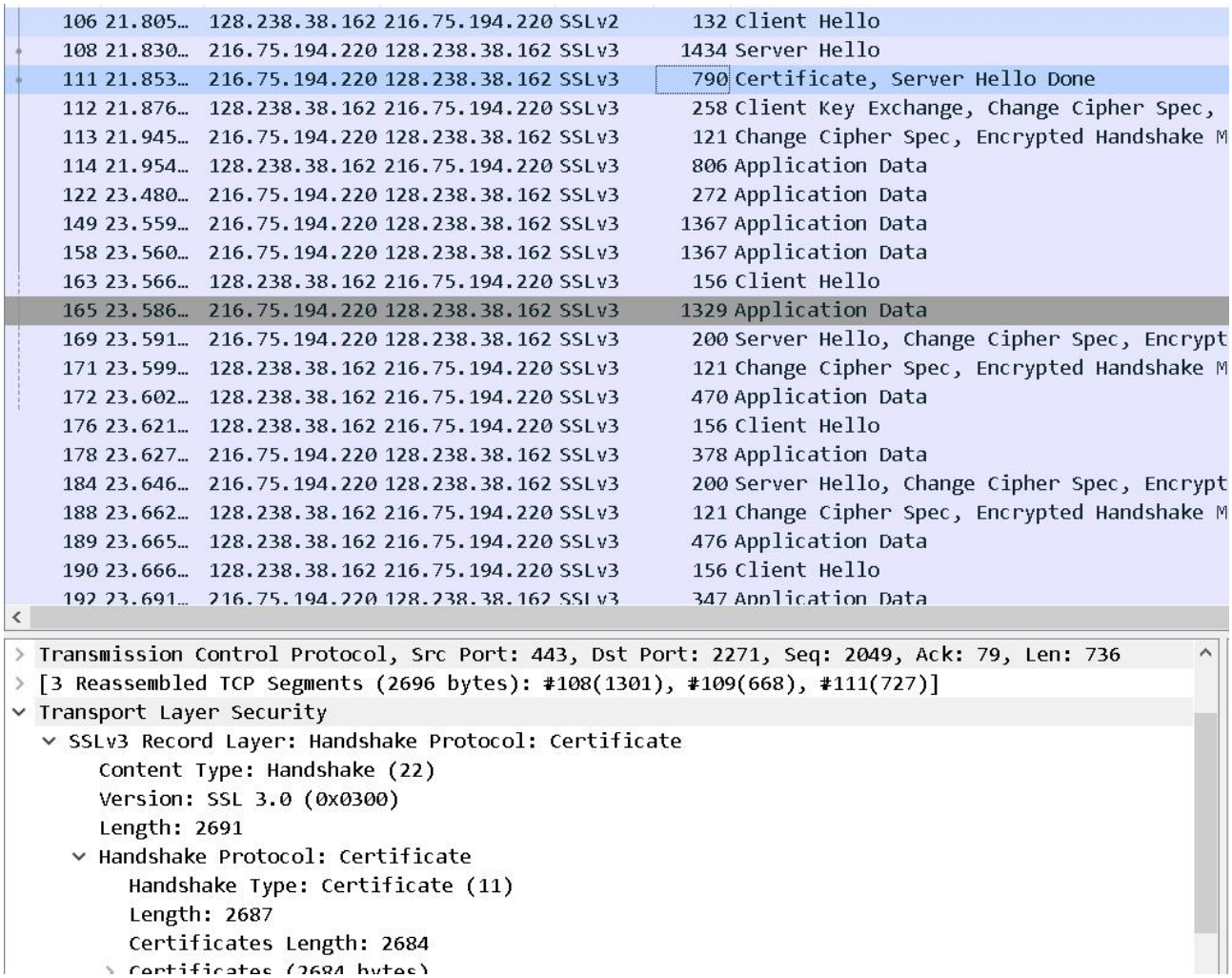


8. 此记录包含了一个 32 字节长的会话 ID，它的目的是在一定时间内，使用会话 ID 连接快速恢复连接过程。用户首次进行登录时需要进行验证，验证成功以后服务器就下发 session id，之后客服端每次请求就携带 session id，就不用在进行验证。



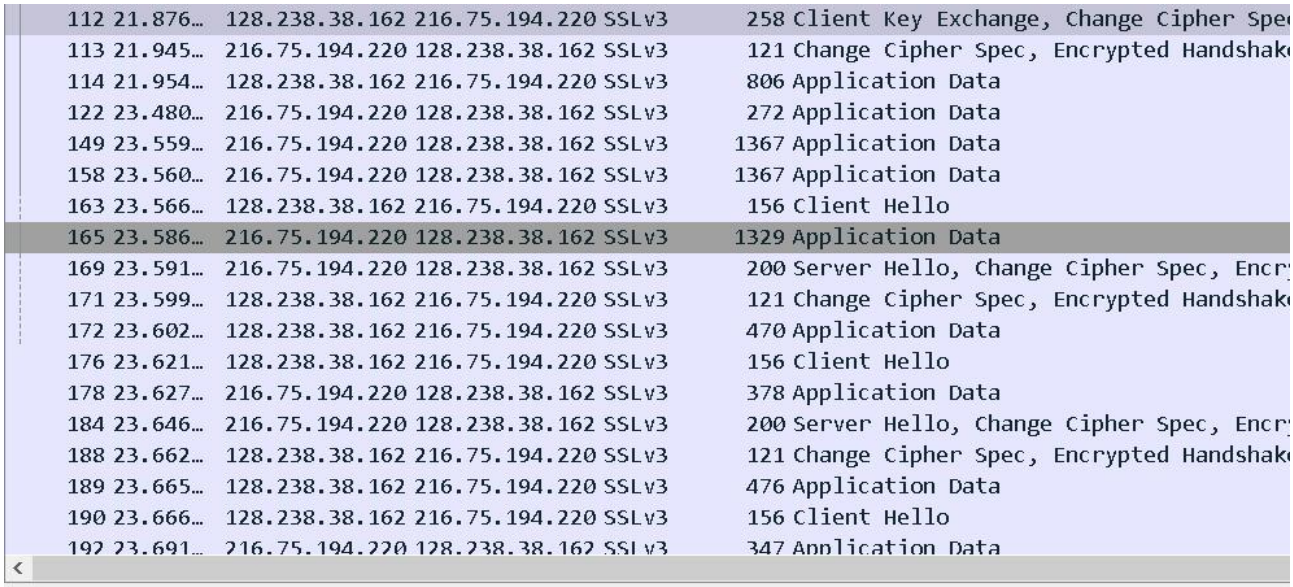
9. 此帧是不包含证书的，后面的帧是包含证书的，适合在一个单独的以太网帧传输。


106 21.805… 128.238.38.162 216.75.194.220 SSLv2 132 Client Hello
108 21.830… 216.75.194.220 128.238.38.162 SSLv3 1434 Server Hello
111 21.853… 216.75.194.220 128.238.38.162 SSLv3 790 Certificate, Server Hello Done
112 21.876… 128.238.38.162 216.75.194.220 SSLv3 258 Client Key Exchange, Change Cipher Spec,
113 21.945… 216.75.194.220 128.238.38.162 SSLv3 121 Change Cipher Spec, Encrypted Handshake M
114 21.954… 128.238.38.162 216.75.194.220 SSLv3 806 Application Data
122 23.480… 216.75.194.220 128.238.38.162 SSLv3 272 Application Data
149 23.559… 216.75.194.220 128.238.38.162 SSLv3 1367 Application Data
158 23.560… 216.75.194.220 128.238.38.162 SSLv3 1367 Application Data
163 23.566… 128.238.38.162 216.75.194.220 SSLv3 156 Client Hello
165 23.586… 216.75.194.220 128.238.38.162 SSLv3 1329 Application Data
169 23.591… 216.75.194.220 128.238.38.162 SSLv3 200 Server Hello, Change Cipher Spec, Encrypt
171 23.599… 128.238.38.162 216.75.194.220 SSLv3 121 Change Cipher Spec, Encrypted Handshake M
172 23.602… 128.238.38.162 216.75.194.220 SSLv3 470 Application Data
176 23.621… 128.238.38.162 216.75.194.220 SSLv3 156 Client Hello
178 23.627… 216.75.194.220 128.238.38.162 SSLv3 378 Application Data
184 23.646… 216.75.194.220 128.238.38.162 SSLv3 200 Server Hello, Change Cipher Spec, Encrypt
188 23.662… 128.238.38.162 216.75.194.220 SSLv3 121 Change Cipher Spec, Encrypted Handshake M
189 23.665… 128.238.38.162 216.75.194.220 SSLv3 476 Application Data
190 23.666… 128.238.38.162 216.75.194.220 SSLv3 156 Client Hello
192 23.691… 216.75.194.220 128.238.38.162 SSLv3 347 Application Data
Transmission Control Protocol, Src Port: 443, Dst Port: 2271, Seq: 2049, Ack: 79, Len: 736
[3 Reassembled TCP Segments (2696 bytes): #108(1301), #109(668), #111(727)]
Transport Layer Security
SSLv3 Record Layer: Handshake Protocol: Certificate
Content Type: Handshake (22)
Version: SSL 3.0 (0x0300)
Length: 2691
Handshake Protocol: Certificate
Handshake Type: Certificate (11)
Length: 2687
Certificates Length: 2684
Certificates (2684 bytes)

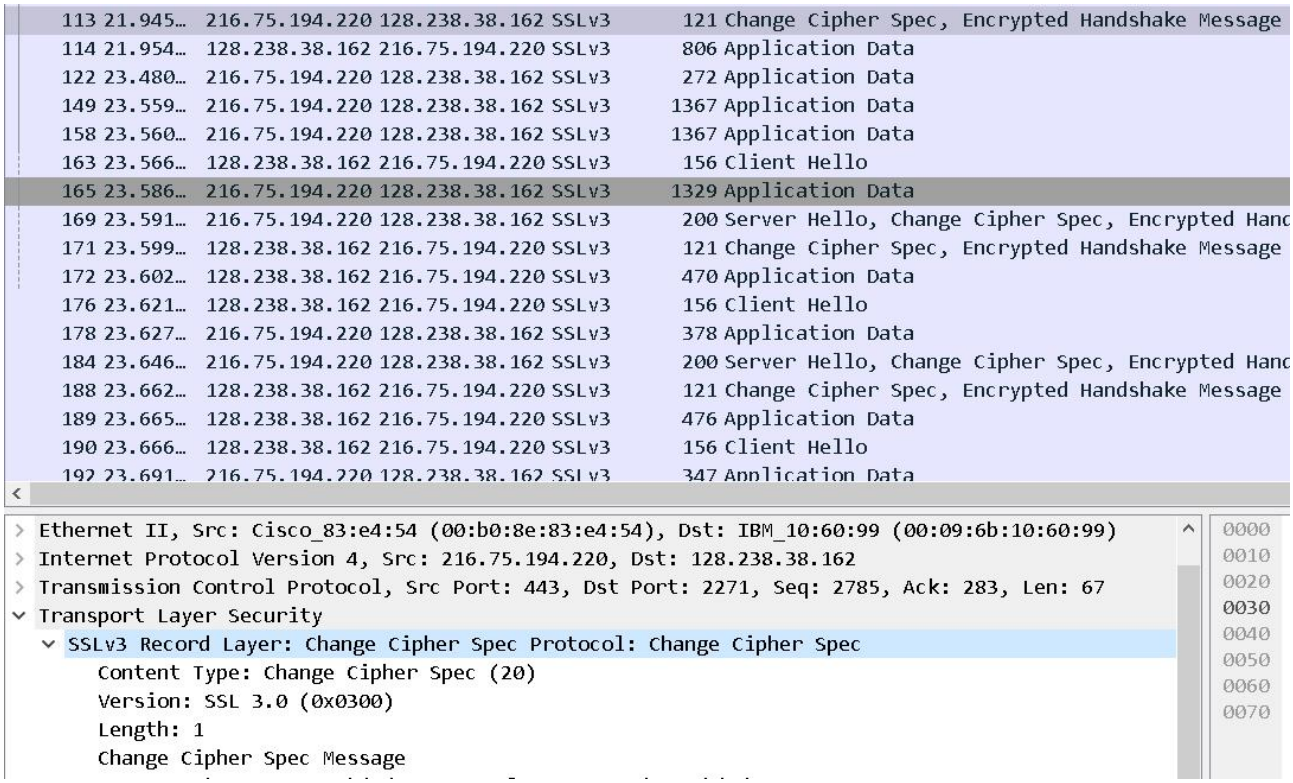


106	21.805…	128.238.38.162	216.75.194.220	SSLv2	132	Client Hello
108	21.830…	216.75.194.220	128.238.38.162	SSLv3	1434	Server Hello
111	21.853…	216.75.194.220	128.238.38.162	SSLv3	790	Certificate, Server Hello Done
112	21.876…	128.238.38.162	216.75.194.220	SSLv3	258	Client Key Exchange, Change Cipher Spec,
113	21.945…	216.75.194.220	128.238.38.162	SSLv3	121	Change Cipher Spec, Encrypted Handshake M
114	21.954…	128.238.38.162	216.75.194.220	SSLv3	806	Application Data
122	23.480…	216.75.194.220	128.238.38.162	SSLv3	272	Application Data
149	23.559…	216.75.194.220	128.238.38.162	SSLv3	1367	Application Data
158	23.560…	216.75.194.220	128.238.38.162	SSLv3	1367	Application Data
163	23.566…	128.238.38.162	216.75.194.220	SSLv3	156	Client Hello
165	23.586…	216.75.194.220	128.238.38.162	SSLv3	1329	Application Data
169	23.591…	216.75.194.220	128.238.38.162	SSLv3	200	Server Hello, Change Cipher Spec, Encrypt
171	23.599…	128.238.38.162	216.75.194.220	SSLv3	121	Change Cipher Spec, Encrypted Handshake M
172	23.602…	128.238.38.162	216.75.194.220	SSLv3	470	Application Data
176	23.621…	128.238.38.162	216.75.194.220	SSLv3	156	Client Hello
178	23.627…	216.75.194.220	128.238.38.162	SSLv3	378	Application Data
184	23.646…	216.75.194.220	128.238.38.162	SSLv3	200	Server Hello, Change Cipher Spec, Encrypt
188	23.662…	128.238.38.162	216.75.194.220	SSLv3	121	Change Cipher Spec, Encrypted Handshake M
189	23.665…	128.238.38.162	216.75.194.220	SSLv3	476	Application Data
190	23.666…	128.238.38.162	216.75.194.220	SSLv3	156	Client Hello
192	23.691…	216.75.194.220	128.238.38.162	SSLv3	347	Application Data

10. 包含，前主密钥用于派生主秘钥，服务器和客户端都用它来生成主密钥，主密钥也是用来进行加密的，长度是 128bytes。

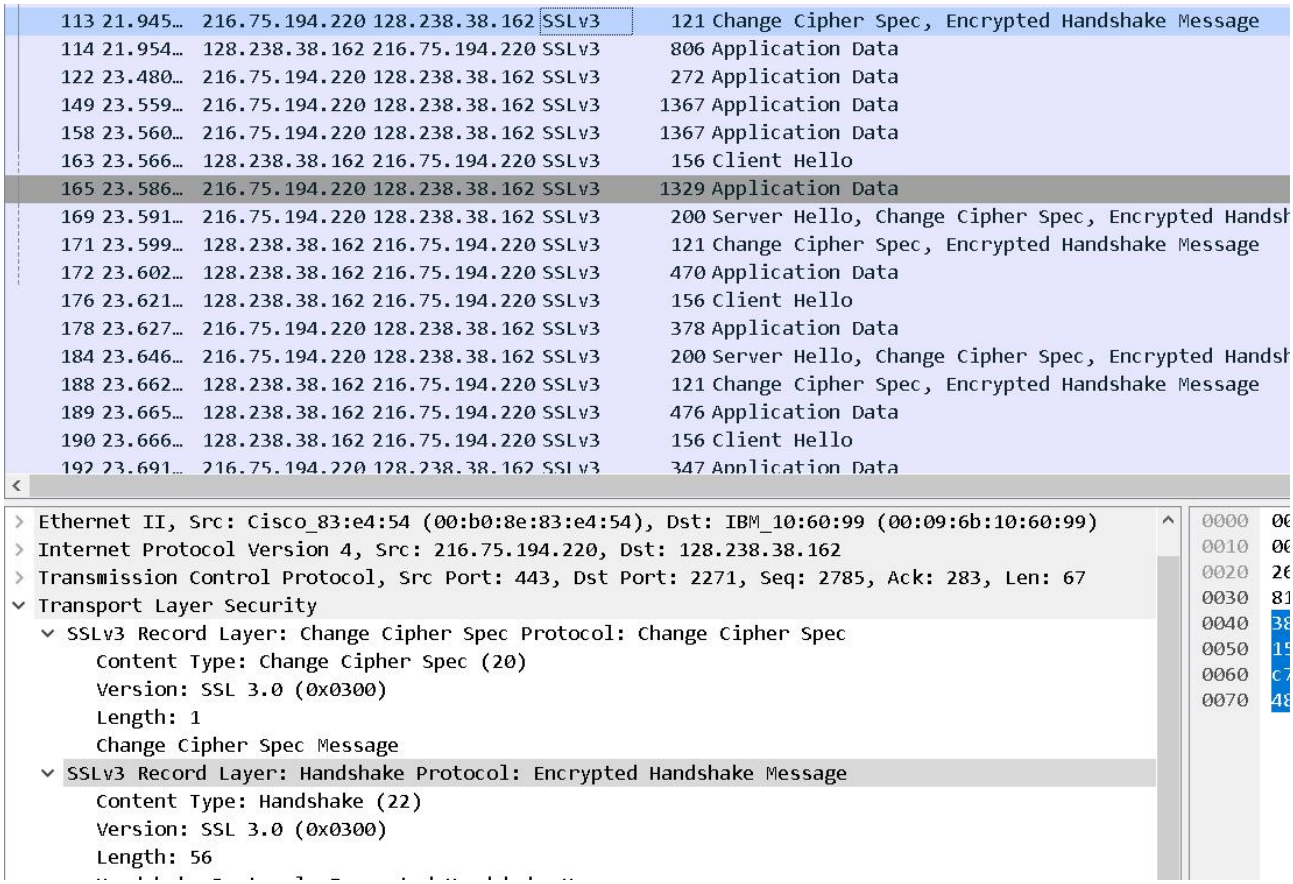


11. 更改密码规范的目的是指示加密和身份验证算法的更改，并在接下来的 SSL 记录中使用，以后将会用商定的加密方式和密钥加密传输，一共有 6 个字节。

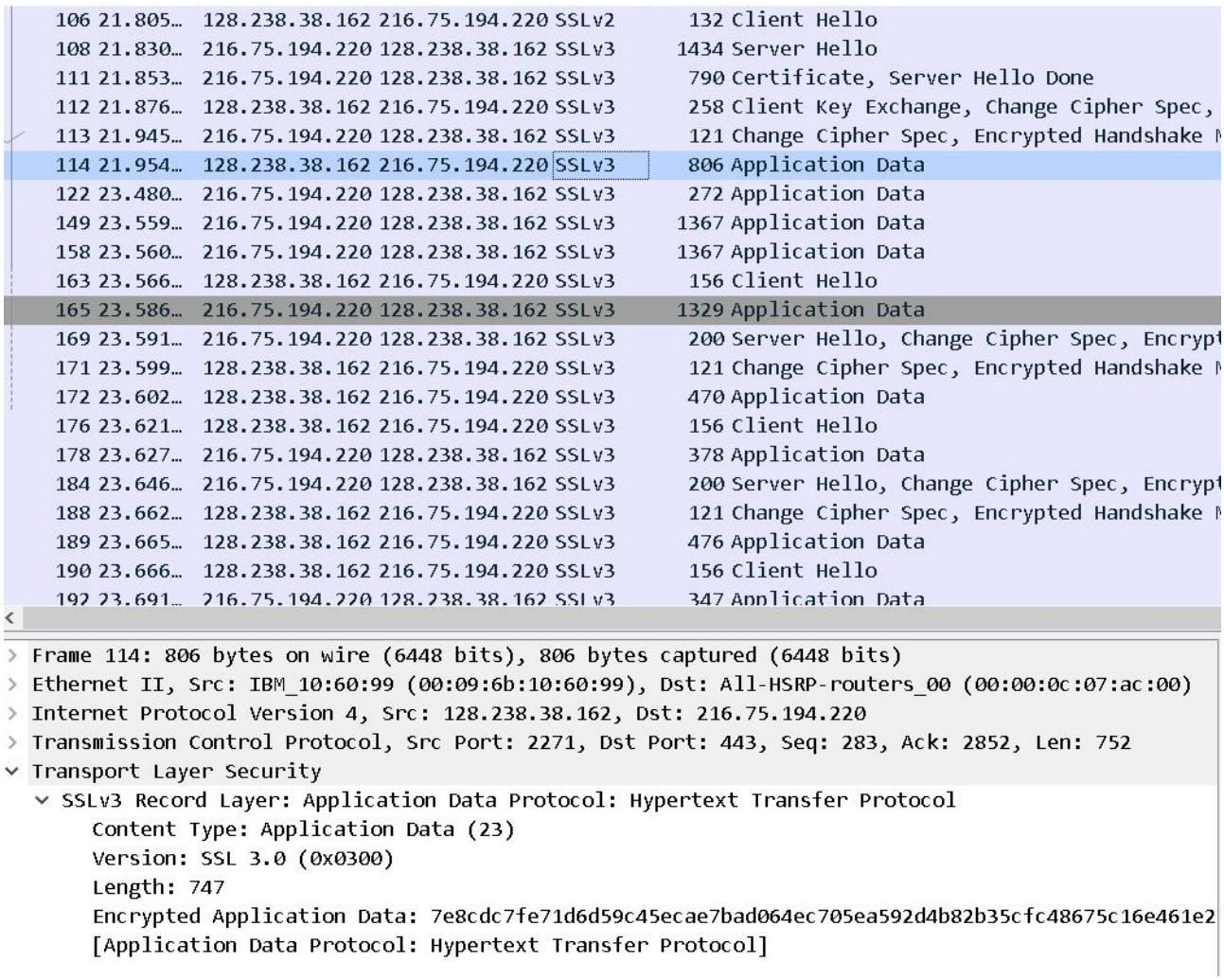


12. 消息校验码是加密的，这个校验码是包含之前所有连接消息的摘要加密格式，只有服务器可以解开，因为在建立连接中，存在可能连接消息被侦听和更改的情况，因此还需要进行信息摘要计算和加密传输，判断是否存在异常，如果异常，将会直接关闭连接。加密握手记录用于验证密钥交换和身份验证过程是否成功。

13. 服务器也会发送更改密码规范记录和加密握手记录到客户端，相同，加密握手记录中同样是包含之前所有连接消息摘要的加密形式，用以供客户端解密，判断是否存在异常选择处理。



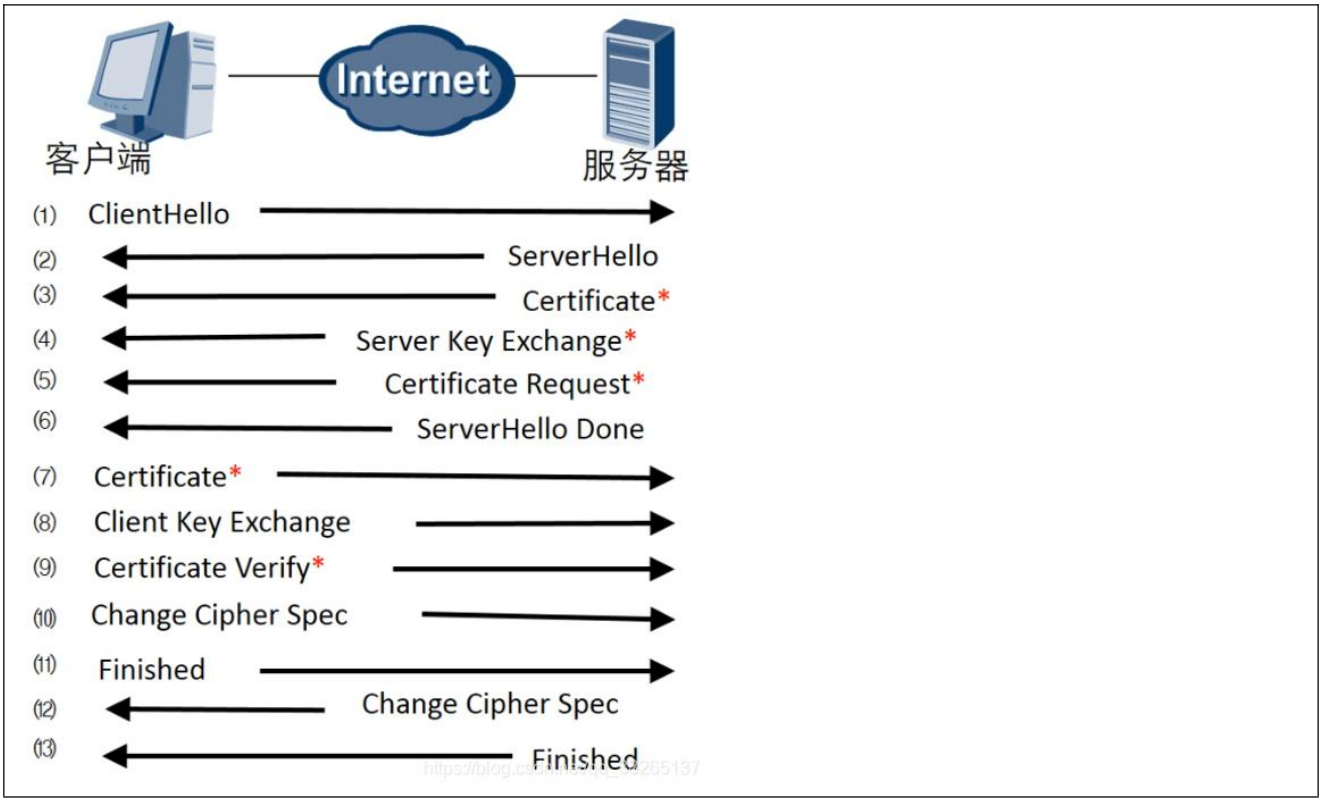
14. 使用本次对话协商和交换好对称加密密钥应用程序数据，包含应用程序数据的记录是包含 MAC，但不区分加密后的应用数据和 MAC。



15. SSL 的版本使用了从最初的 ClientHello 消息中的 SSLv2 到之后所以消息交换中的 SSLv3 的更改。

结论分析与体会：

SSL 同时使用了对称加密和非对称加密，不直接使用 RSA 对数据进行加密是因为使用 RSA 进行加密的话所需计算量较大。在本实验中先从获取服务端的证书和服务端的公钥，之后客户端通过公钥加密前主密钥发给服务端，此时客户端和服务端都有主密钥，即可进行加密传输。


Internet
客户端
服务器
(1) ClientHello
(2) ServerHello
(3) Certificate*
(4) Server Key Exchange*
(5) Certificate Request*
(6) ServerHello Done
(7) Certificate*
(8) Client Key Exchange
(9) Certificate Verify*
(10) Change Cipher Spec
(11) Finished
(12) Change Cipher Spec
(13) Finished