Nova Southeastern University

College of Computing and Engineering

**Assignment 3**

**3ISEC 660 Advanced Network Security**

Winter 2021

Due date: 3/14/2021

Total Points: 100

Q1. Suppose someone suggests the following way to confirm that the two of you are both in possession of the same secret key. You create a random bit string the length of the key, XOR it with the key, and send the result over the channel. Your partner XORs the incoming block with the key (which should be the same as your key) and sends it back. You check, and if what you receive is your original random string, you have verified that your partner has the same secret key, yet neither of you has ever transmitted the key. Is there a flaw in this scheme? Please use examples(s) to justify your answer.

Yes, there is a known flaw in this scheme. If an attacker listens in on the transmission, they can acquire both the originators string value of its key and random string, along with the repliers string value of the key and the originators string mentioned above. Because of that they attacker will be able to solve the key by solving the Originators String XOR the Repliers String.

Example:

1.Originator generates random character string R.

2.OrigString = (Key XOR R)

3.Send OrigString to Replier.

4.Replier calculates ReplyString. (ReplyString = OrigString XOR Key)

5.Replier sends ReplyString to Originator.

6.Originator verifies if R is equal to ReplyString or not.

If an attacker is eavesdropping on the network and acquires the OrigString and ReplyString, they will be able to solve the key via Key =(OrigString XOR ReplyString)

Q2. Consider the RSA algorithm with *p=7* and *q=13*. Follow the conventions shown in Section 21.4.

2.1 What are *n* and ∅(𝑛)?

**N** = P\*Q so that N= (7\*13)

=> **N=91**

**∅(𝑛)** = (P-1)\*(Q-1) so that (6\*12)

=> **∅(𝑛)=72**

2.2 Let *e* be 3. Is this an acceptable choice for *e*? If the answer is *yes*, then justify. If the answer is *no*, suggest an alternative value for *e*.

**No e=3 is NOT an acceptable value**, because although 1< 3 < 72 is true, the greatest common divisor of 3 and 72 is 3 and not 1.

**A number that does satisfy is e=5** because 1<5<72 and the GCD of 5 and 72 is 1 which satisfies the need for e to be acceptable.

2.3 Find *d* such that *de*=1 (mod *z*) and *d*<72.

Since an acceptable value of e is 5, will use e=5.

de mod z = 1 so that dk\*5=1mod72.

d= (72k+1)/5 = 73/5 = 14.6 (Won’t work because not a whole number.)

d= (72\*2k+1)/5 = 145/5 = 29 (Is a whole number so satisfies need.)

**d = 29**

d. What are public key and private key in this question?

The public key is **(n=91,e=5).**

The private key is **(n=91,d=29)**

Q3. Cryptography (Main Reference: Chapter 21)

Review Section 21.5 of the textbook on the Diffie-Hellman algorithm. Following the textbook convention, with 𝑞=71 and 𝛼=7, suppose user A and B choose private keys 𝑋𝐴=5 and 𝑋𝐵=12, respectively.

3.1 Calculate A's and B's public keys, 𝑌𝐴 and 𝑌𝐵. Show the process.

𝑌𝐴 = 7\*5mod71 **= 51**

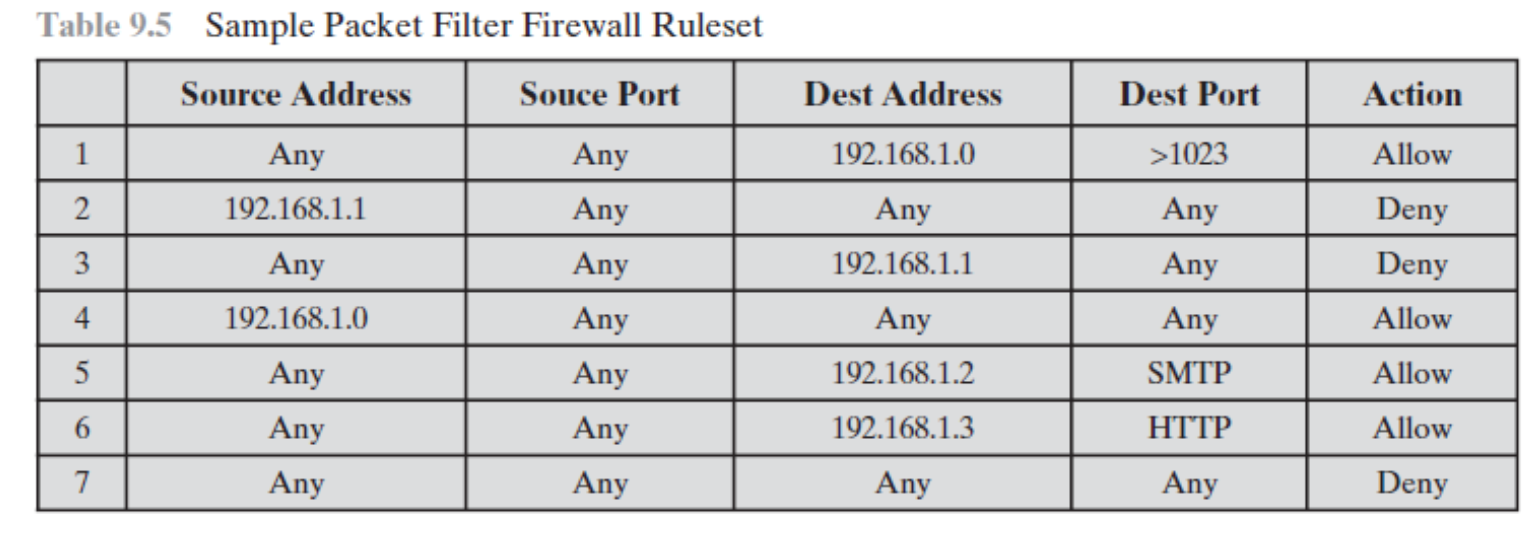
𝑌𝐵 = 7\*12 mod 71 **= 4**

3.2 Following the results from step a, calculate the shared symmetric key K.

𝑌K = 4\*5mod71 **= 30**

Q4. Firewall (Main Reference: Chapter 9)

Table 9.5 of the textbook shows a sample of a packet filter firewall rule set for an imaginary network of IP address that range from 192.168.1.0 to 192.168.1.254. Describe the effect of each firewall rule.



1: Everyone can send packets to 192.168.1.0 on ports over the number 1023.

2: 192.168.1.1 cannot send packets.

3: No packets can be sent to 192.168.1.1.

4: 192.168.1.0 can send packets to anyone.

5: Anyone can send packets to 192.168.1.2 using SMTP.

6: Anyone can send packets to 192.168.1.3 using HTTP.

7: Explicit deny all.

Q5. List four functions supported by S/MIME.

The four functions supported by S/MIME are:

Authentication – Verify user is who they say they are.

Non-repudiation of Origin – Proof of origin, creator can’t deny they sent it.

Message integrity – Proof the original message sent is what was received.

Message privacy – Added security that message is encrypted and can’t be read.

Q6. What are the two ways of providing authentications in IPsec?

IPsec offers two modes of authentication. One for the IP packets header and one for the IP packets content.

Authentication Headers(AH) – Provides security to the IP packets header but does not provide security to the contents of the packet.

Encapsulating Security Payload (ESP) –Provides security to the content, but not the header.

Q7. What are the principal elements of a Kerberos system? Why the system is designed with different servers?

The Kerberos protocol has three parts: Application Server, Client Machines, and a Key Distribution Center (KDC) composed of an Authentication Service(AS).

Client makes a request for a ticket to the KDC that uses an AS.

The Client sends the Ticket Granting Ticket(TGT) to the Ticket Granting Service (TGS).

The TGS sends the ticket to access the application server to the client.

The Client sends the ticket to access the application server, and the server responds with a confirmation of its identity.

It is designed with different servers because the KDC essentially acts as a trusted third party between clients and servers as a liaison to mediate trust.

**Section III. Practical assignment (30 points)**

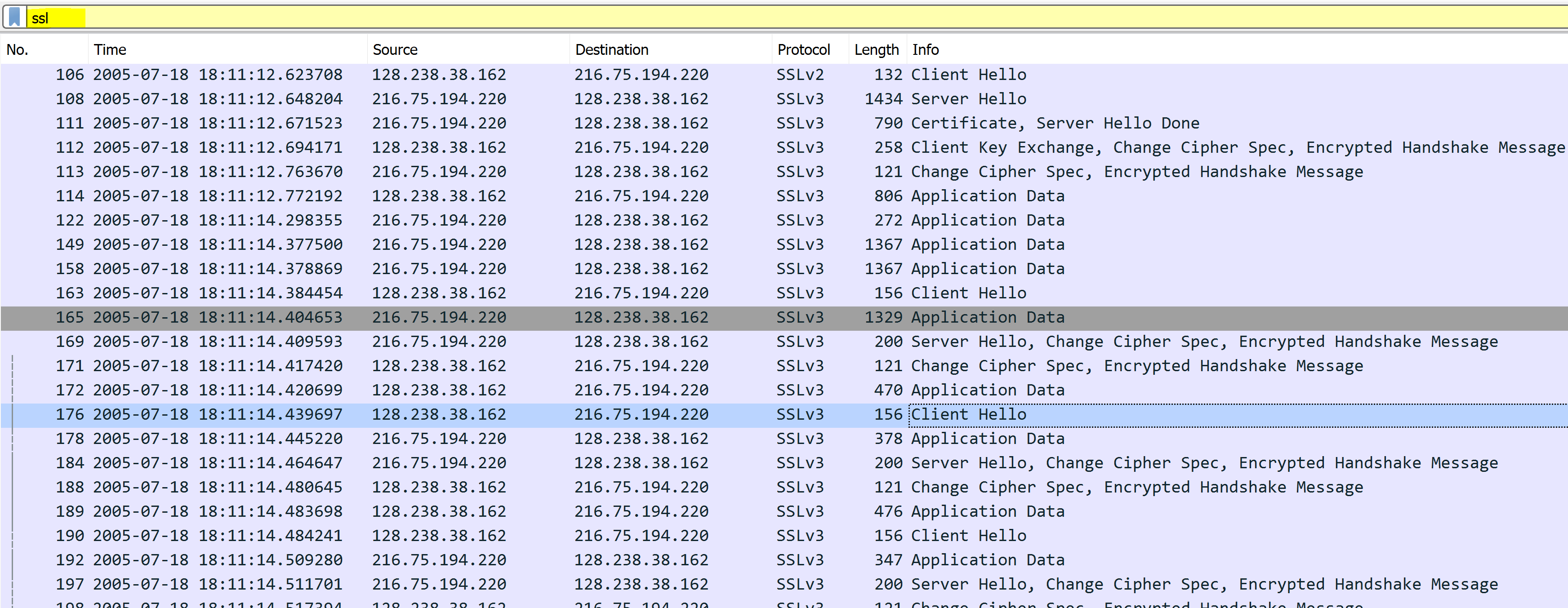
The first step is to capture the packets in an SSL session. To do this, you should go to

your favorite e-commerce site and begin the process of purchasing an item (but

terminating before making the actual purpose!). After capturing the packets with

Wireshark, you should set the filter so that it displays only the Ethernet frames that

contain SSL records sent from and received by your host.



A look at the captured trace

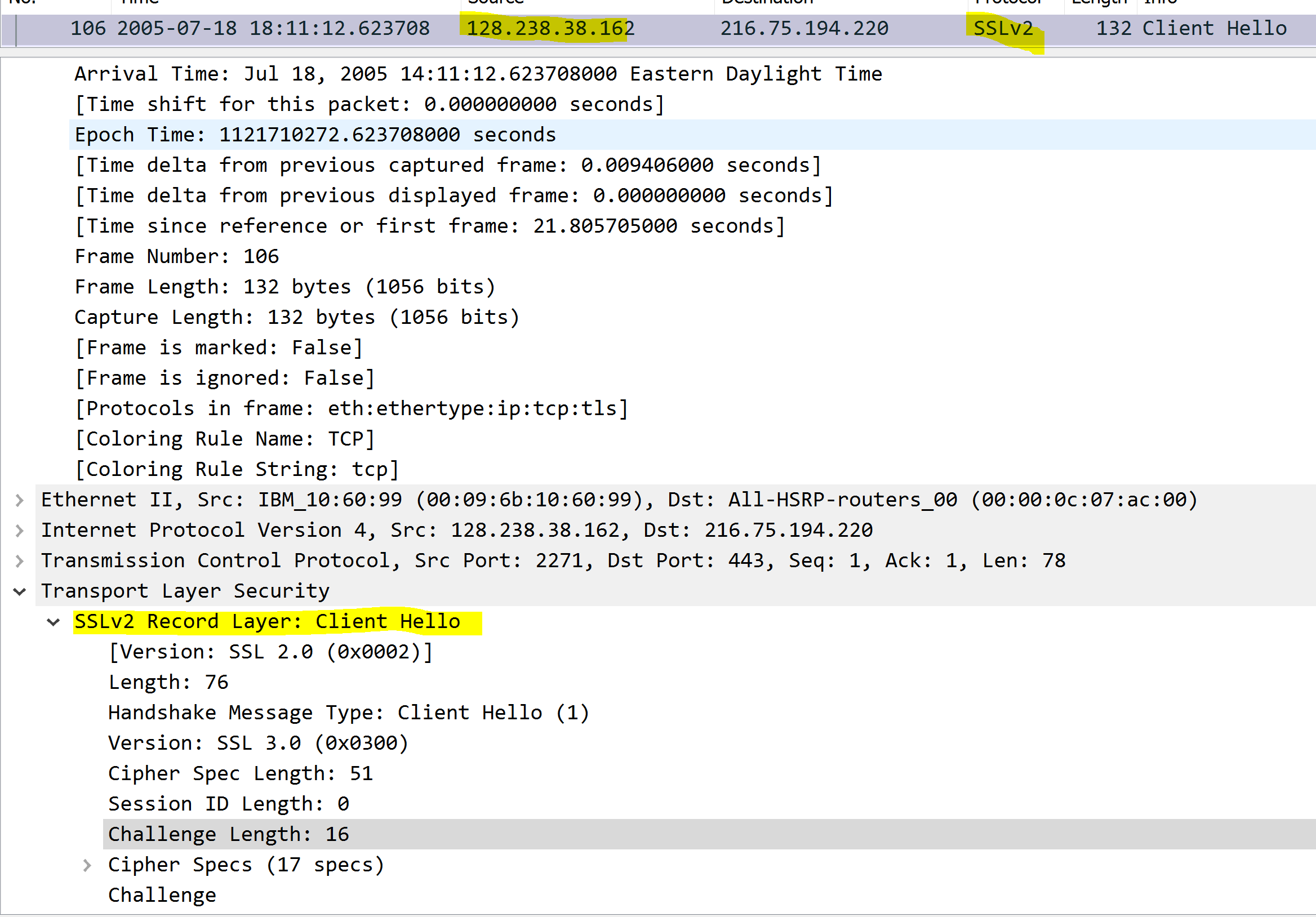
1. For each of the first 8 Ethernet frames, specify the source of the frame (client or

server), determine the number of SSL records that are included in the frame, and

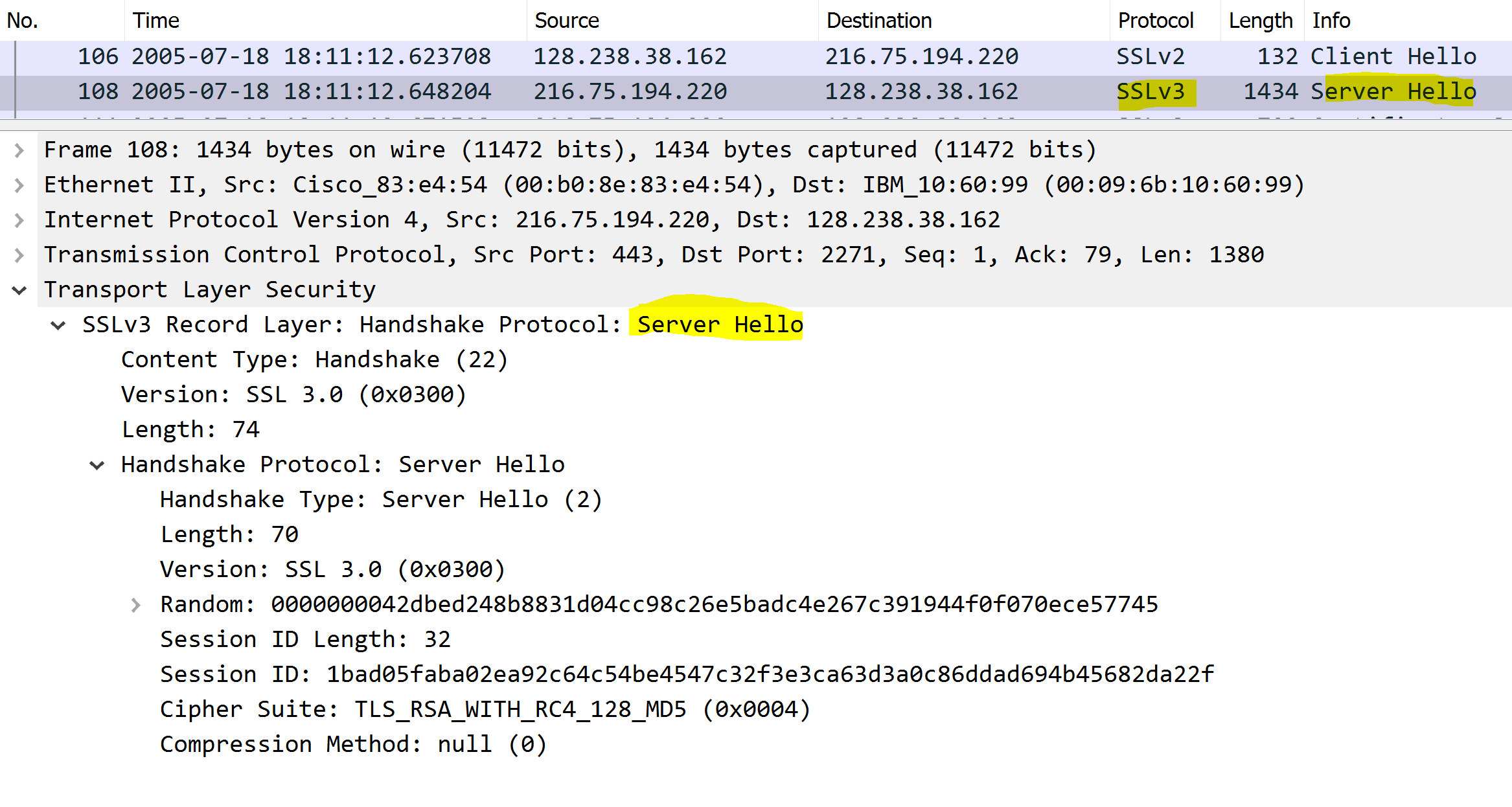
list the SSL record types that are included in the frame. Draw a timing diagram

between client and server, with one arrow for each SSL record.

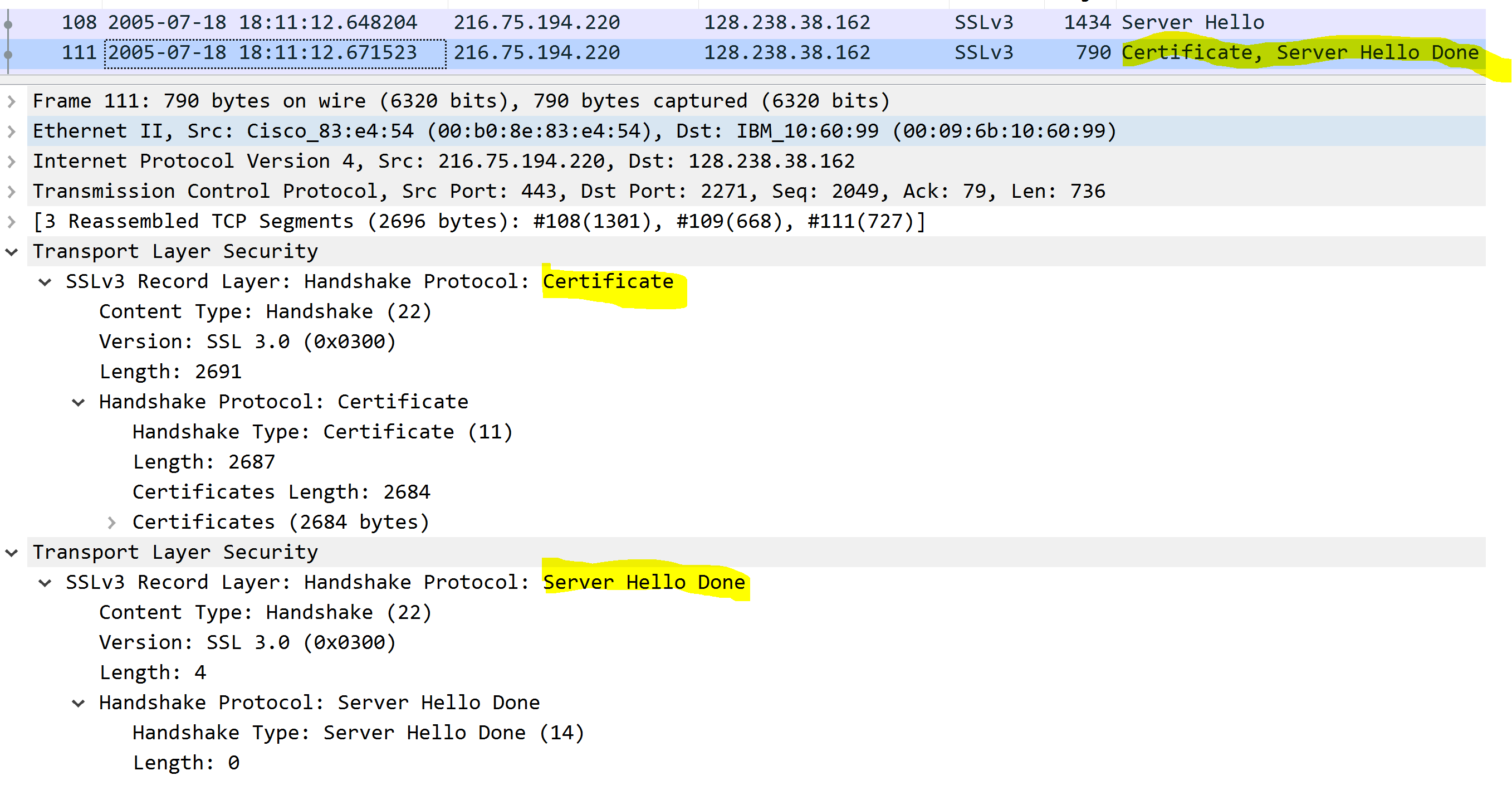
Frame 1: Client sends first SSL record Client Hello.



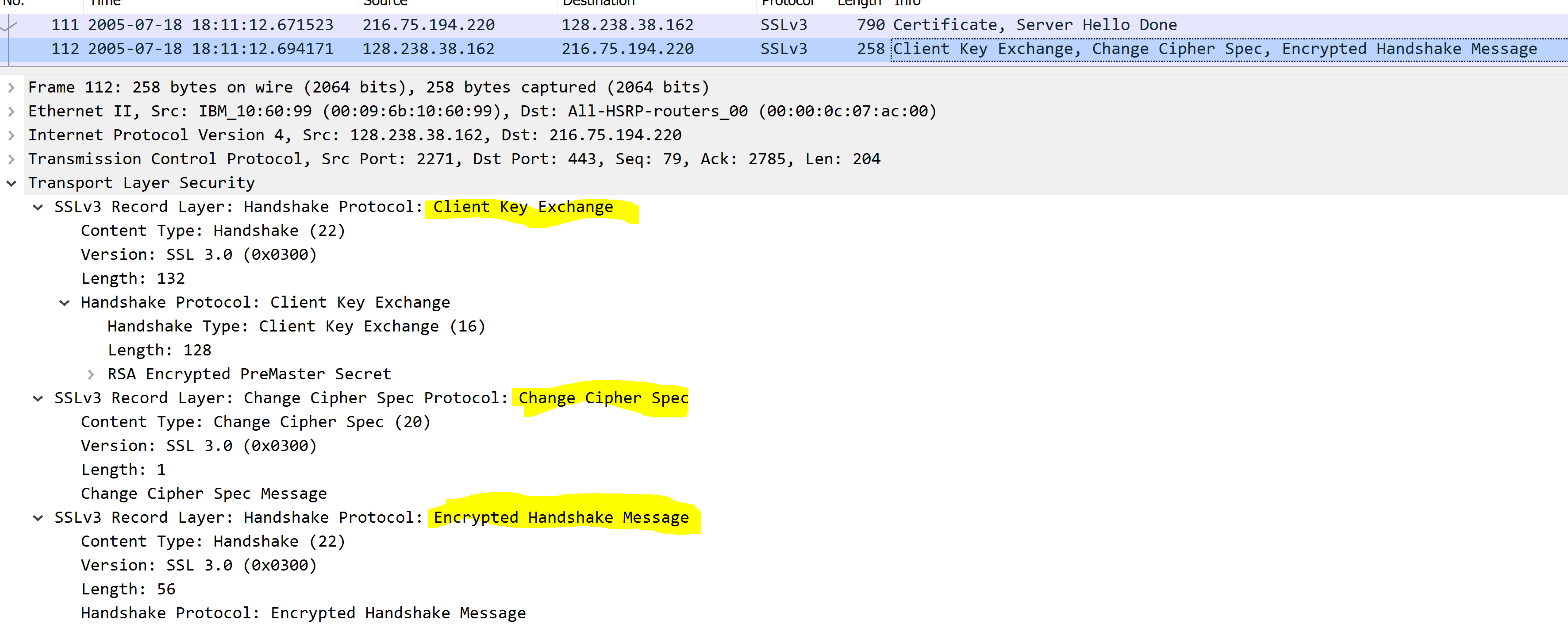
Frame 2: Server responds with Server Hello.



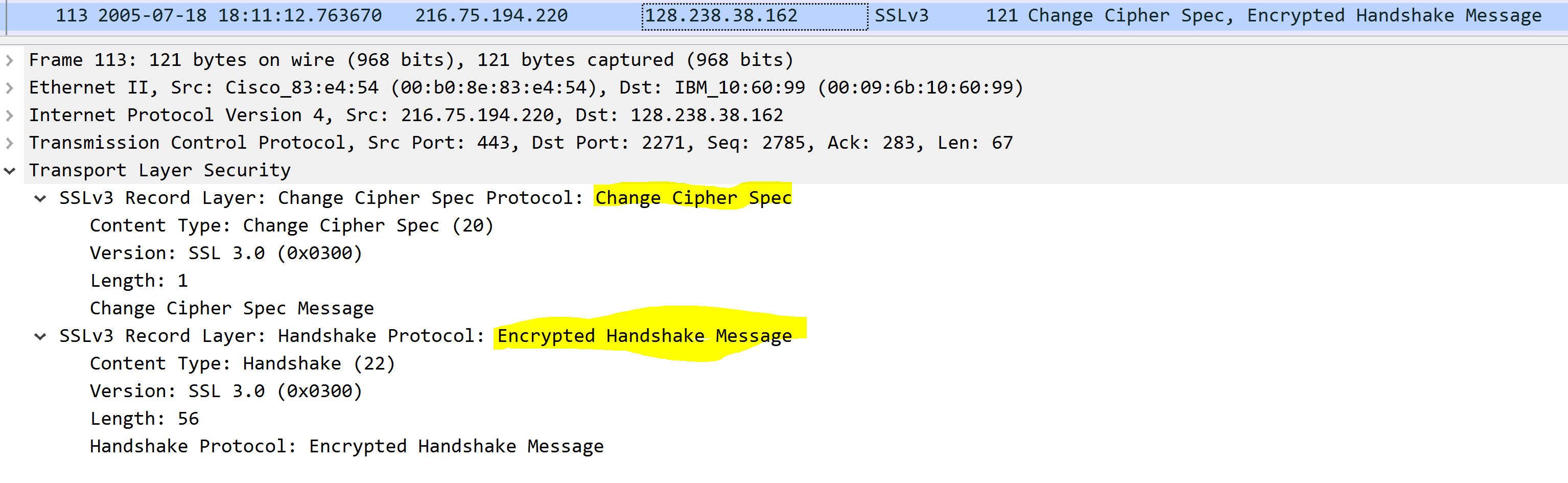
Frame 3: Server Continues to send Certificate to client and Hello Done as the two record types



Frame 4:Client sends key exchange, Change Cipher Spec, and Encrypted Handshake Message.



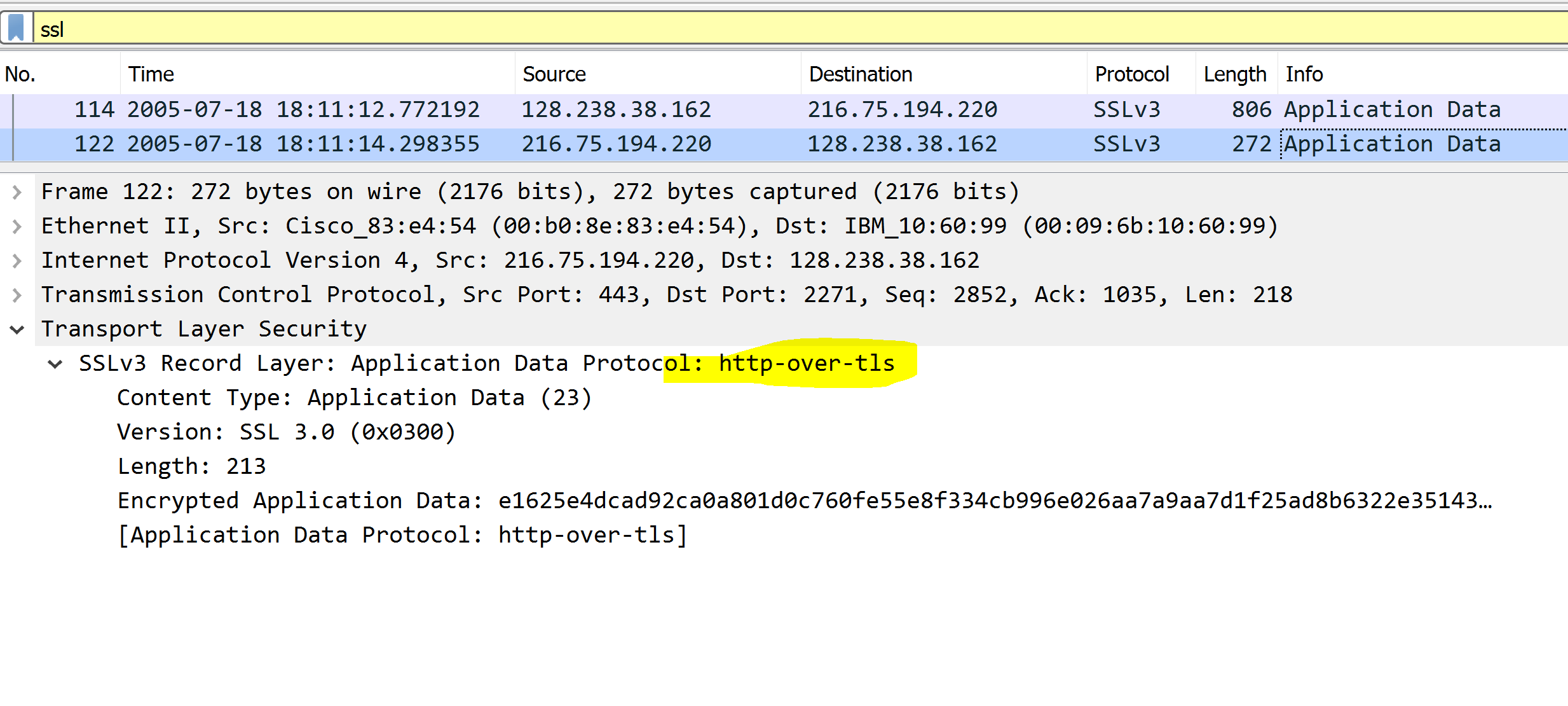
Frame 5:Server responds with received Change Cipher Spec and Encrypted Handshake Message.



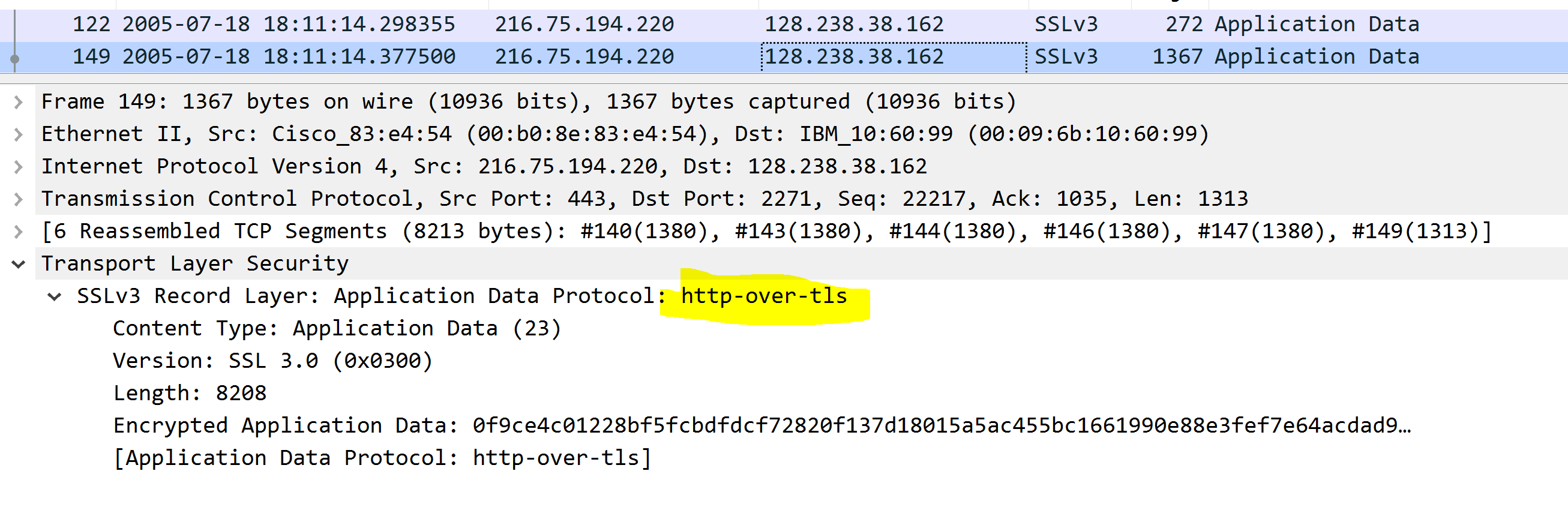
Frame 6:Client begins http-over-tls.



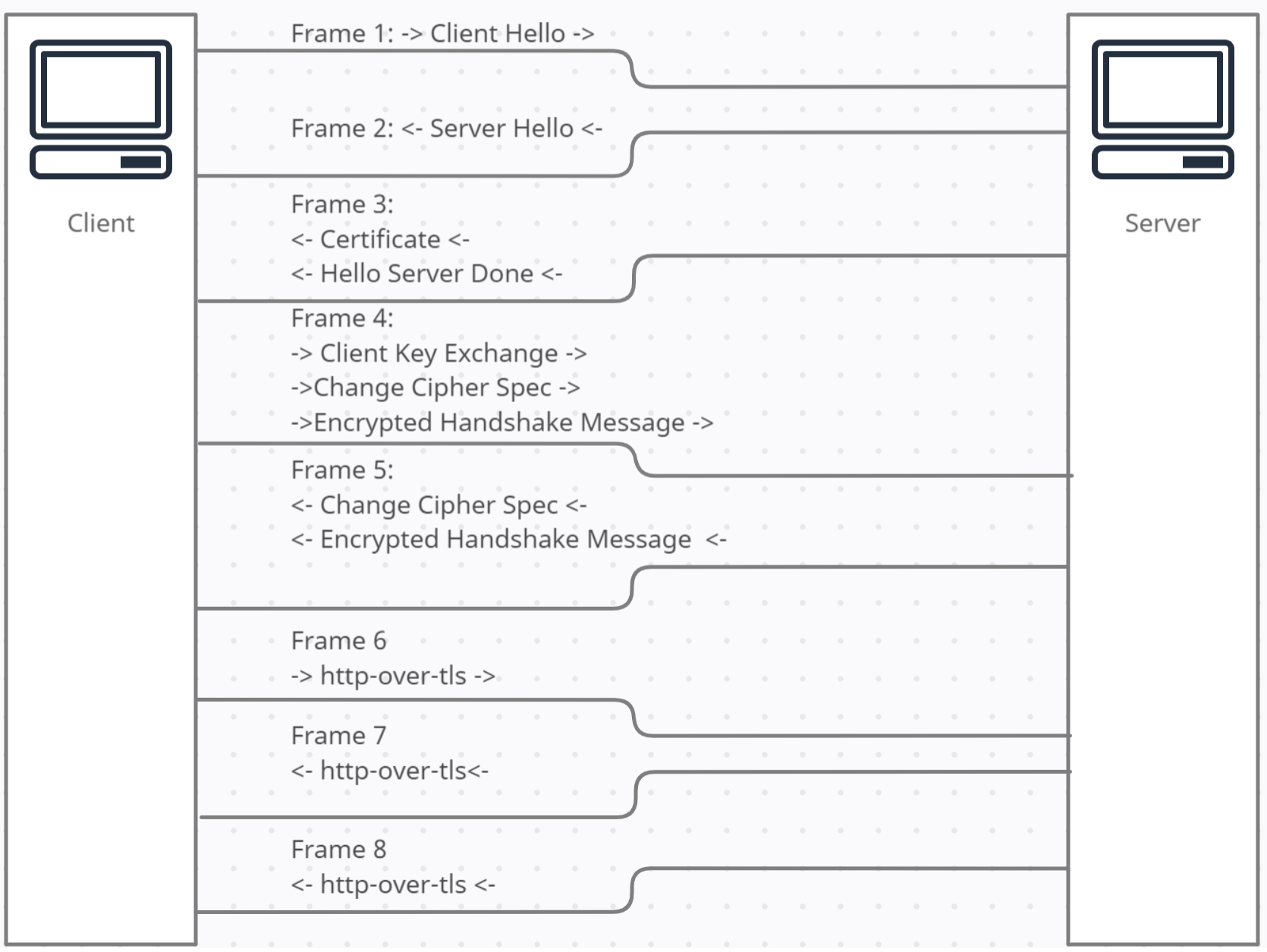
Frame 7:Server sends traffic back with http-over-tls.



Frame 8: Server continues to send traffic via http-over-tls



Timing Diagram



2. Each of the SSL records begins with the same three fields (with possibly different

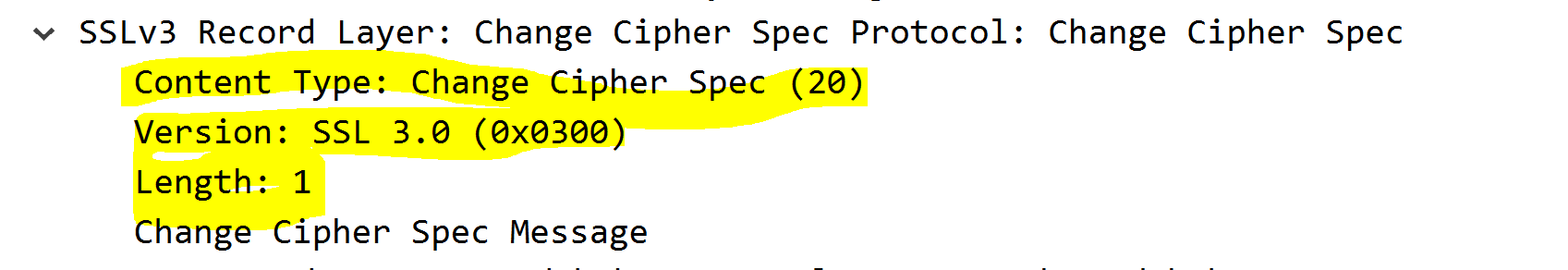
values). One of these fields is “content type” and has length of one byte. List all

three fields and their lengths.

Content Type = 20

Version = (0x0300)

Length: 1



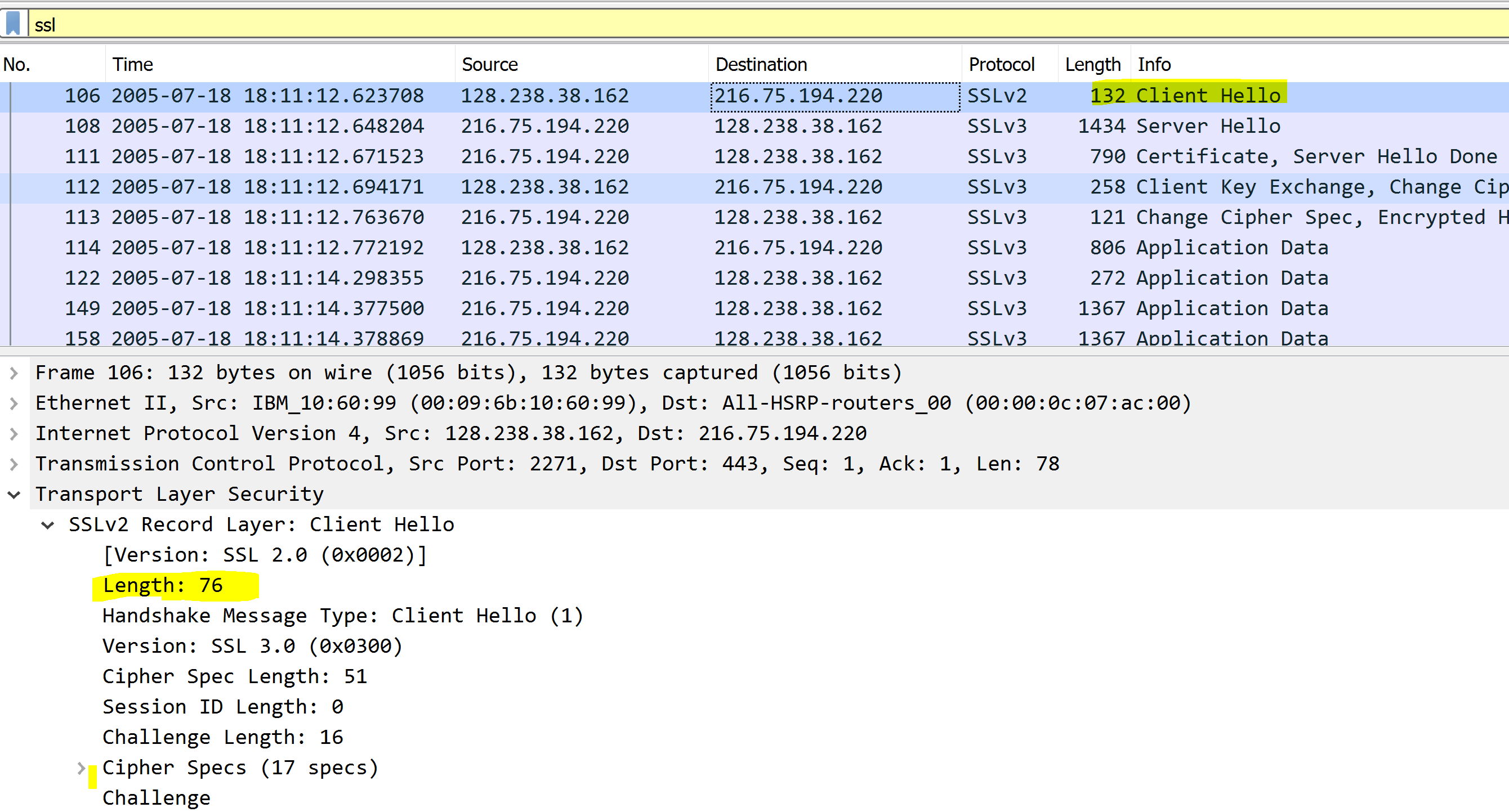
ClientHello Record:

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?

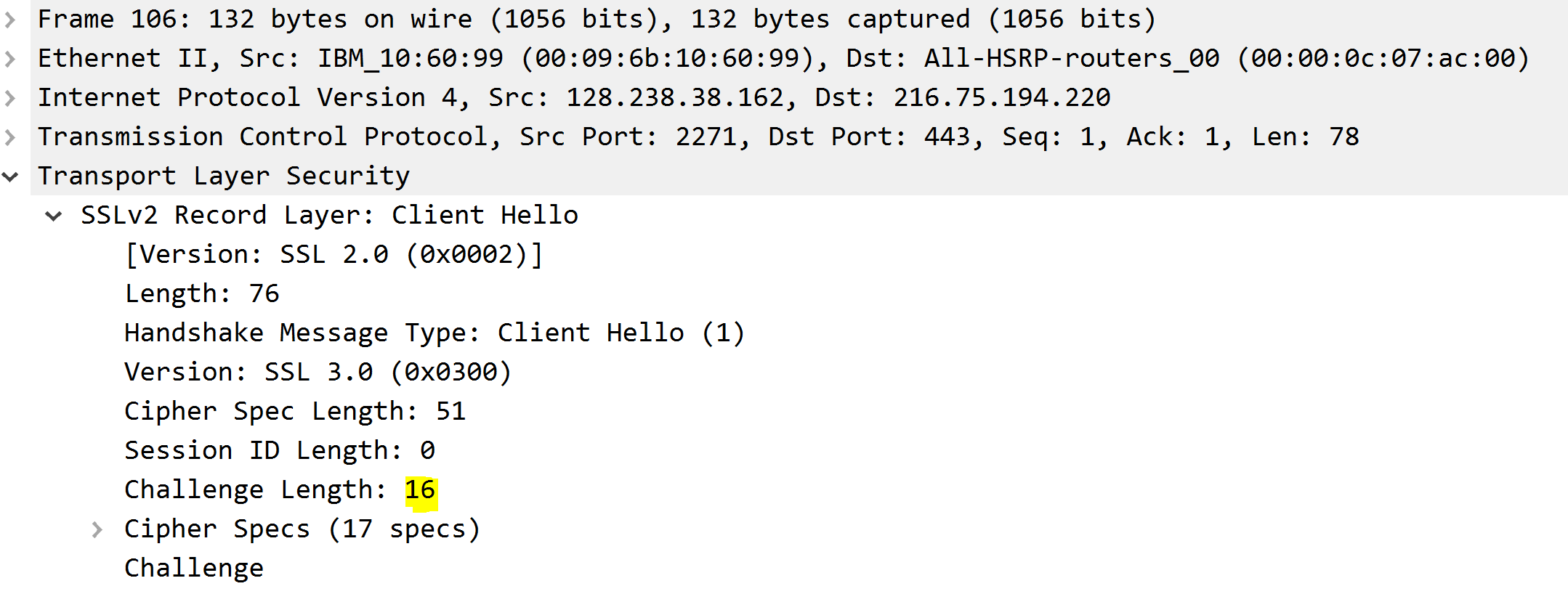
The length is 76 is the ClientHello Record.



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?

Challenge value is 16. In hexadecimal code that would be 10, because the “1” would equal 15 plus the “0” being the first value 1. So that, 15+1 = 16

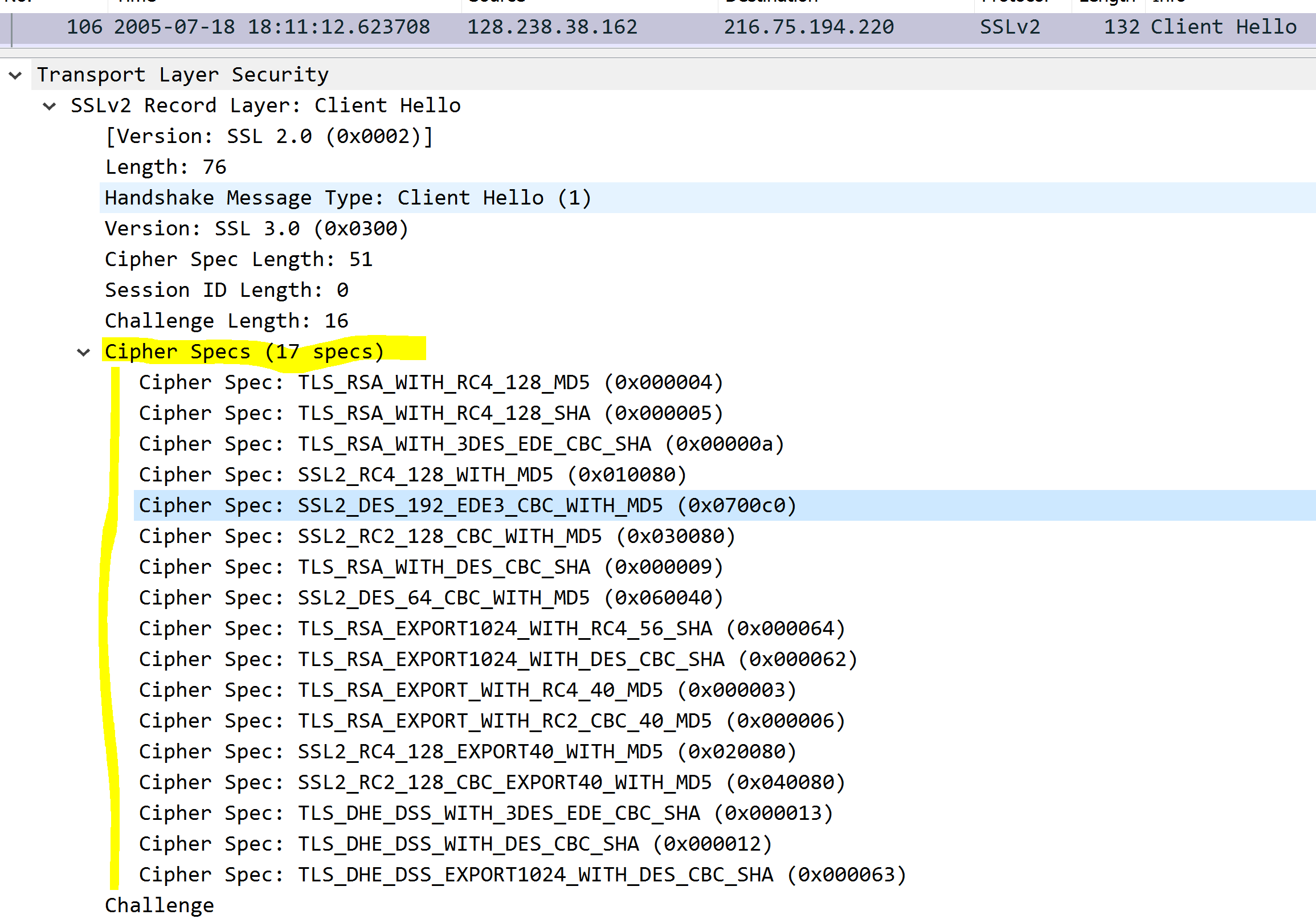


5. Does the ClientHello record advertise the cyber suites it supports? If so, in the

first listed suite, what are the public-key algorithm, the symmetric-key algorithm,

and the hash algorithm?

Yes, it uses RSA for public-key algorithm, it uses RC4 for symmetric-key, and it uses MD5 for hash algorithm.

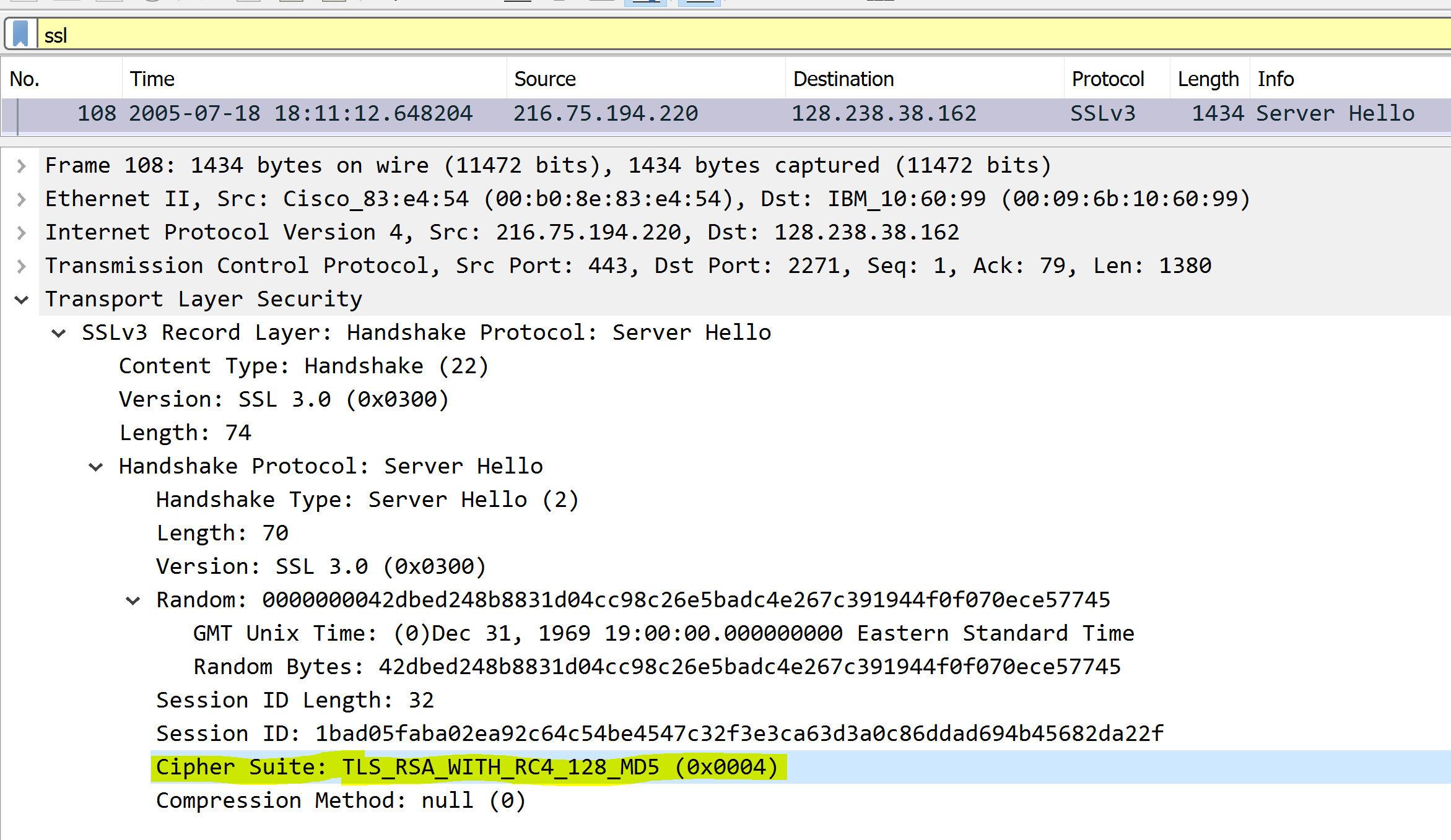


ServerHello Record:

6. Locate the ServerHello SSL record. Does this record specify a chosen cipher

suite? What are the algorithms in the chosen cipher suite?

Yes it uses the Cipher Suite: TLS\_RSA\_WITH\_RC4\_128\_MD5 (0x0004)

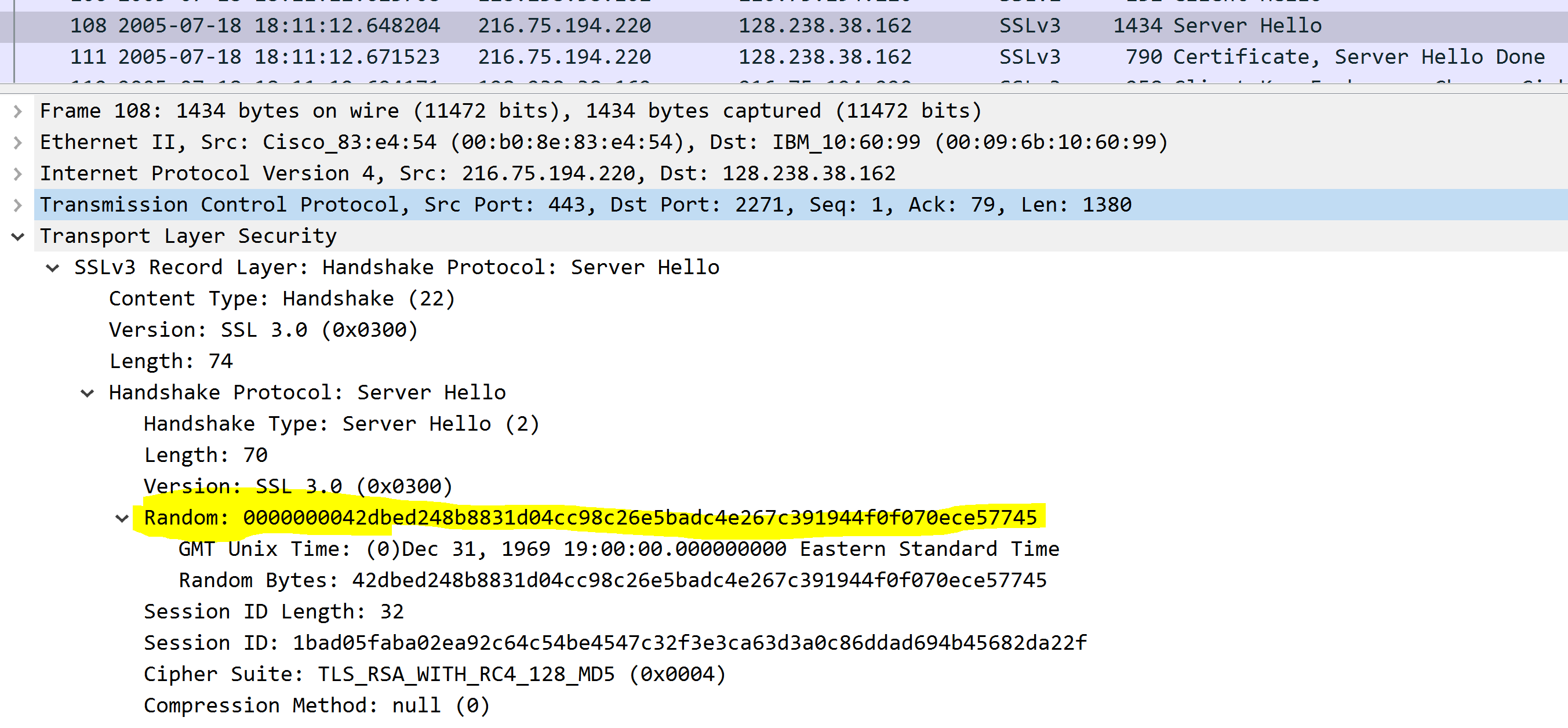


7. Does this record include a nonce? If so, how long is it? What is the purpose of the

client and server nonces in SSL?

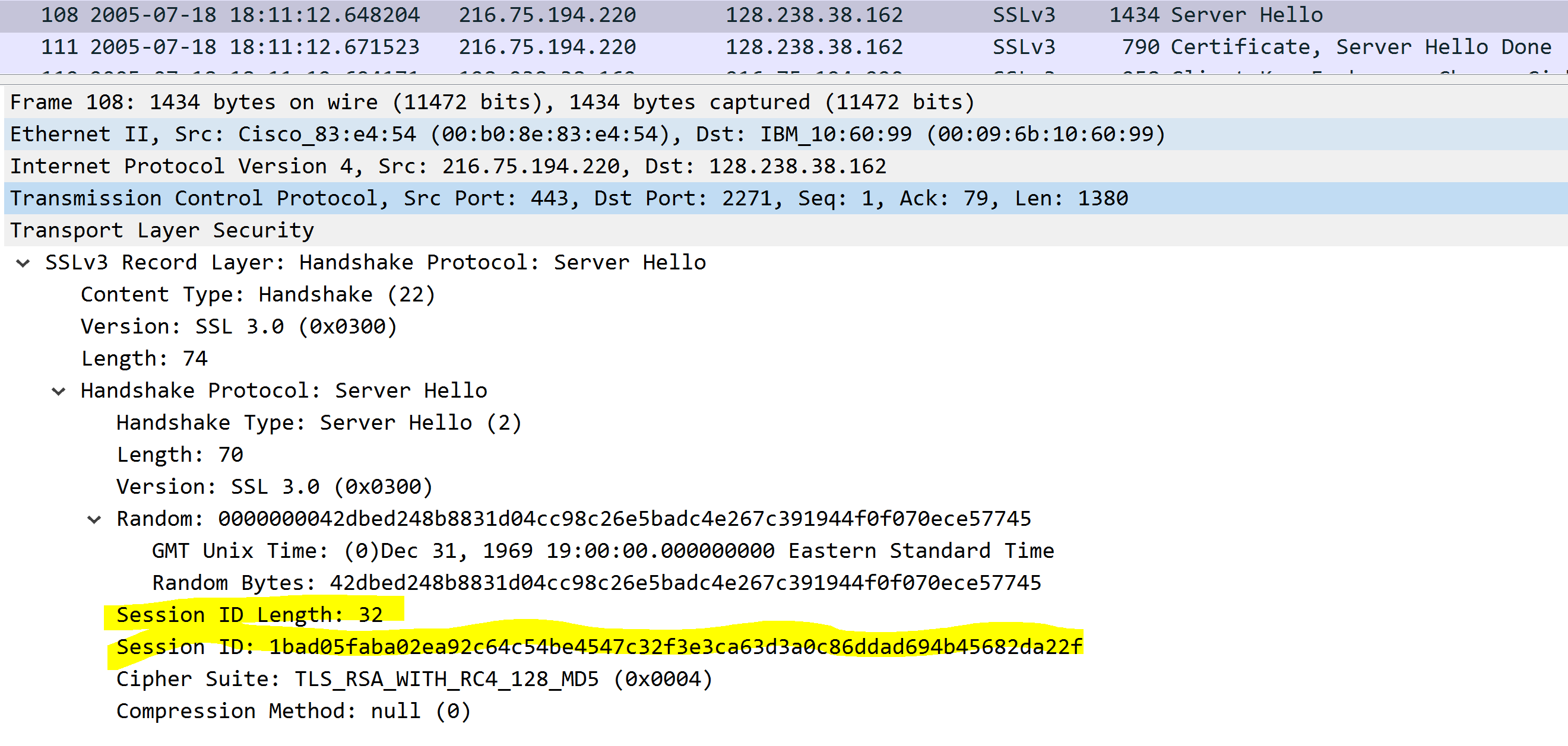
Yes, the nonce is labeled Random. The nonce is 32 bits long, 28 for data and 4

for the time. This prevents a replay attack.



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

Yes, the session ID is exposed with a length of 32. This is used to persist a connection between client and server. The client can use this to resume the same session later.



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?

This record does not provide the certificate in this frame but does in the next, the certificate is small enough that it can fit in a single frame.



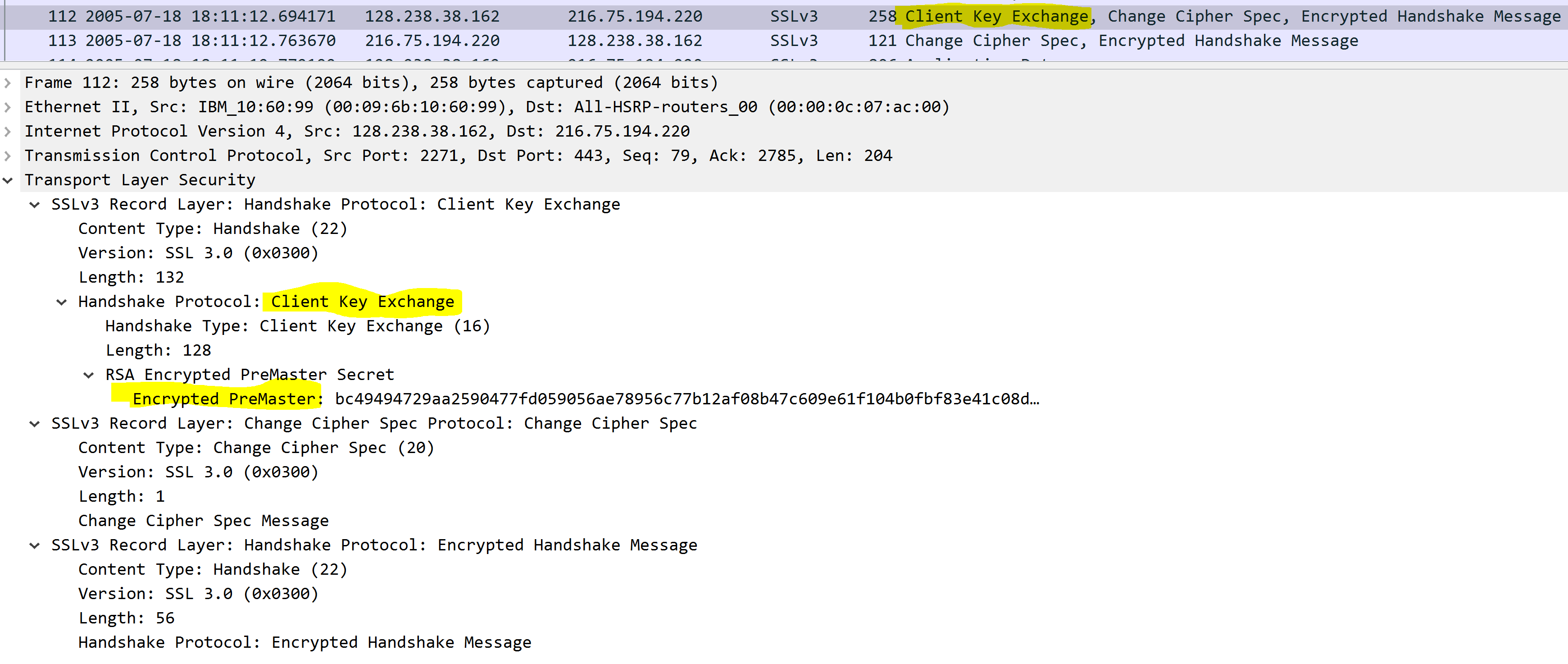
Client Key Exchange Record:

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?

Yes, it does have pre-master secret. This is used to create a master secret to generate session keys. It is 128 bits long.

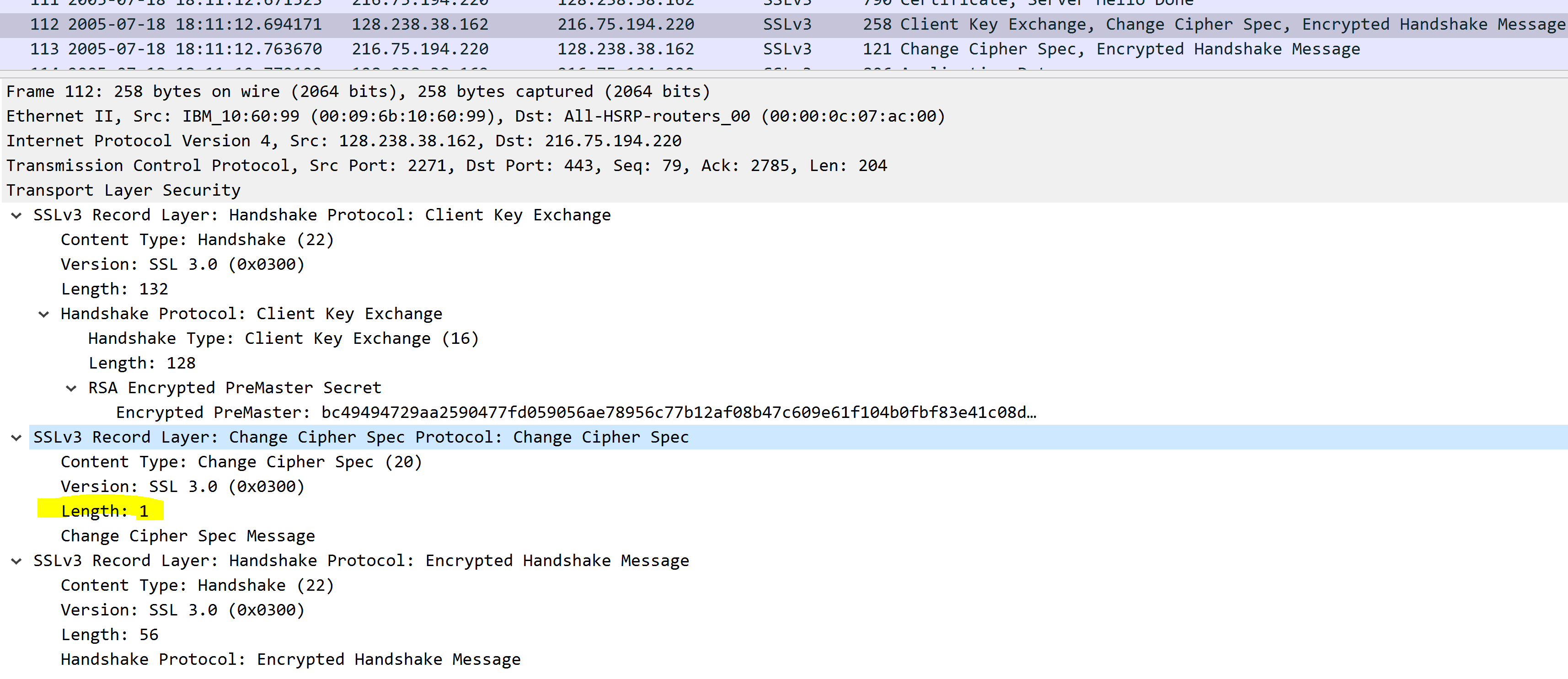


Change Cipher Spec Record (sent by client) and Encrypted Handshake Record:

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

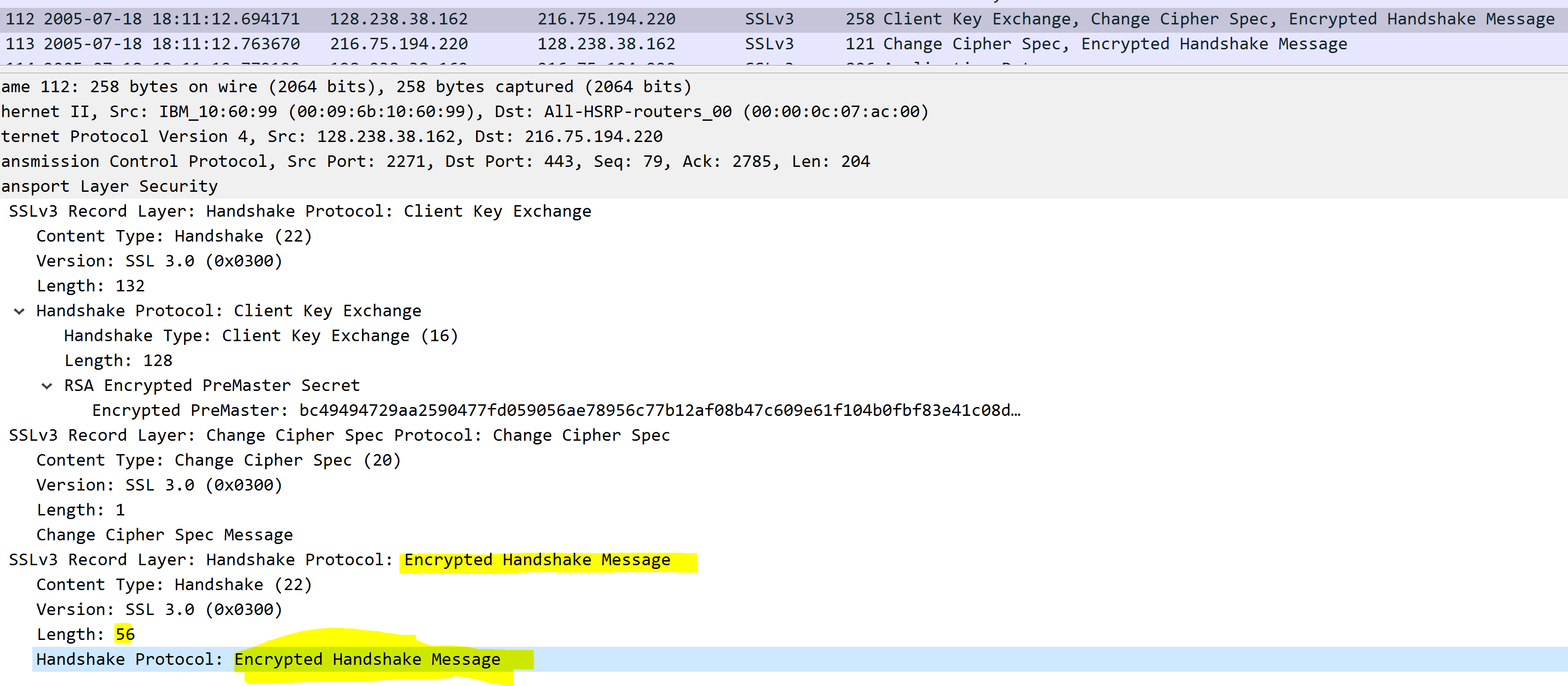
record in your trace?

This is used by the server to verify that the client will send the message contents that are encrypted. The length of bytes in this record trace is 1.



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

In this record a MAC of the concatenation of the all the previous handshakes messages are sent to the server.

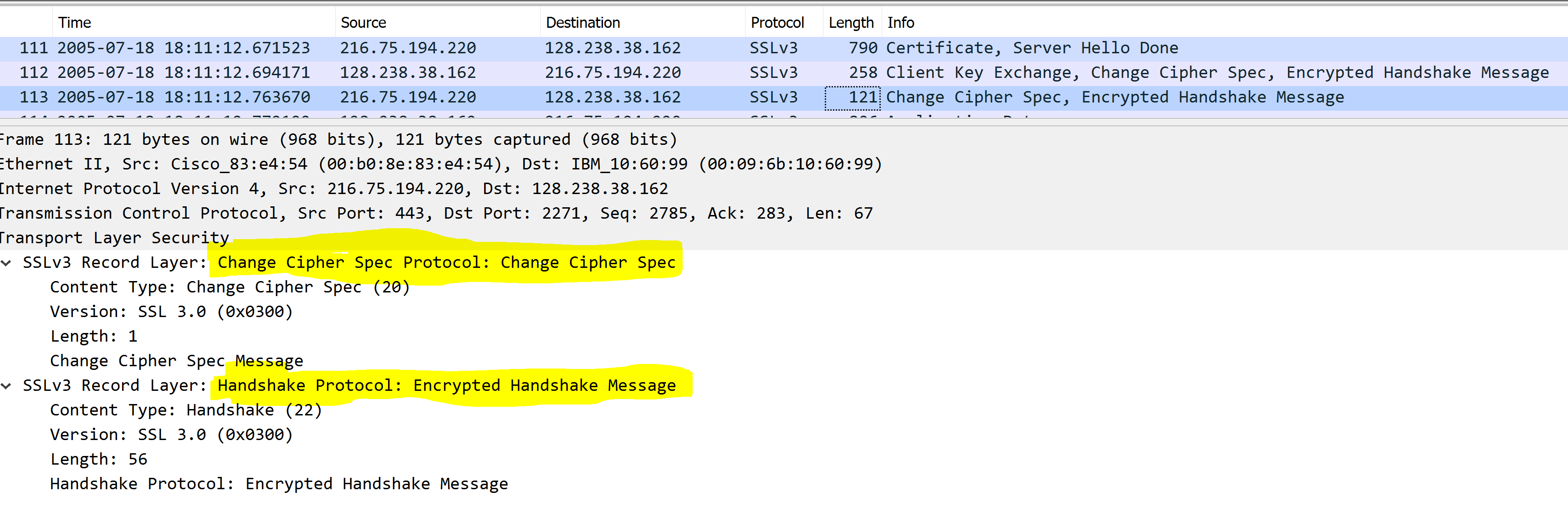


13. Does the server also send a change cipher record and an encrypted handshake

record to the client? How are those records different from those sent by the client?

Yes, the server responses with a change cipher record and encrypted handshake.

Yes the encrypted handshake is different because it is based off all the previous ones so being the next hop up its encrypted message turns out different.



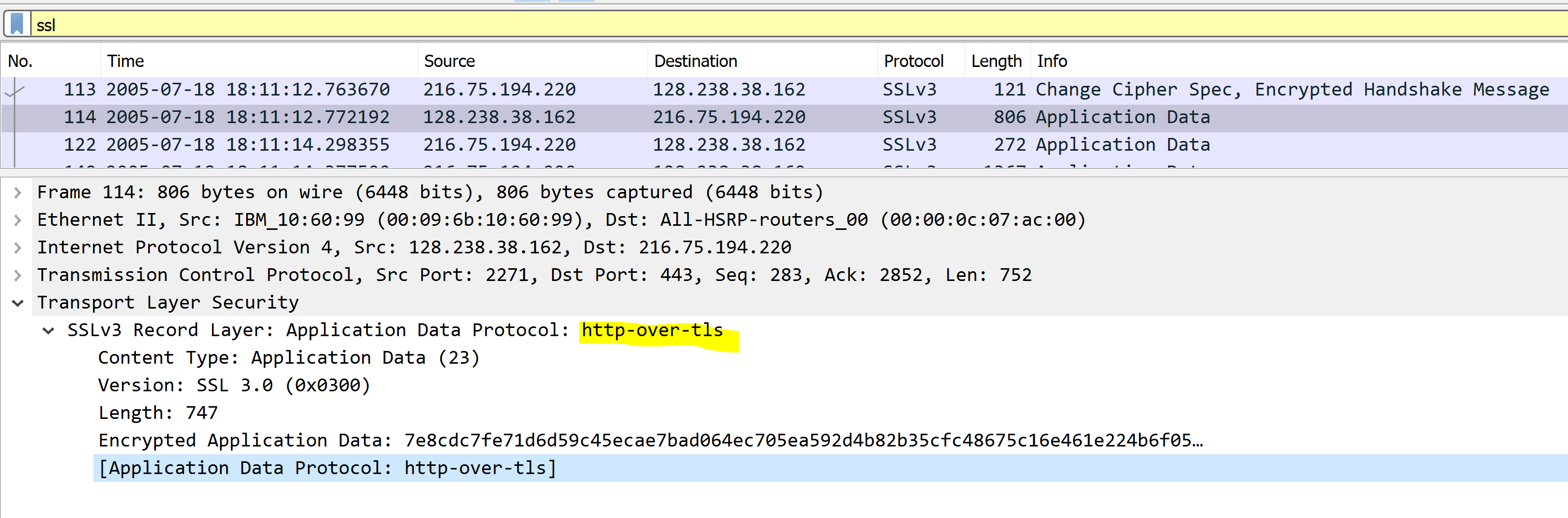
Application Data

14. How is the application data being encrypted? Do the records containing

application data include a MAC? Does Wireshark distinguish between the

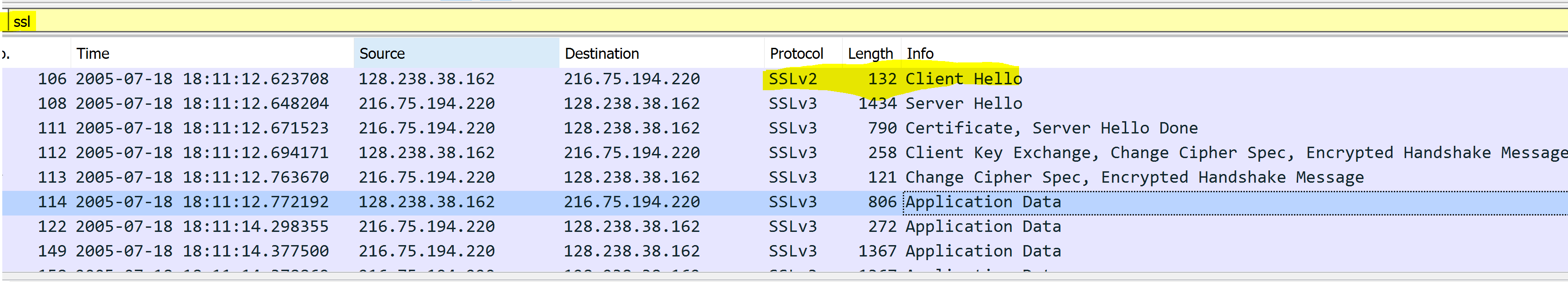
encrypted application data and the MAC?

The application data being encrypted with http-over-tls. Wireshark sees the MAC and the encrypted application data.



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

One thing I found interesting is that the client initialization of the SSL connection begins with SSLv2 before switching of to SSLv3 with the server response.





Certification of Authorship of Assignment

Submitted to (Professor’s Name): Dr. Wei Li

Class/Semester: ISEC660 Advanced Network Security, Winter 2021.

Students’ Names: Eric Webb

Date of Assignment: 3-14-2021

Title of Assignment: Assignment #3

Certification of Authorship: By submitting this document we certify that we are the authors of this paper and that any assistance we received in its preparation is fully acknowledged and disclosed in the paper. We have also cited any sources from which we used data, ideas or words, either quoted directly or paraphrased. We also certify that this paper was prepared by us specifically for this course.

Students’ Signatures: Eric Webb