A Little SSL/TLS

detailyang@gmail.com
2017/03/09

SSL 是什么

是一种安全协议， 全称 Secure Sockets Layer, 网景公司（Netscape）在1994年推出,在 Web 上做为一种安全传输协议。

TLS 是什么

也是一种安全协议， 全称是 Transport Layer Security, 1999 年 RFC 2246 提出，基于 SSL 3.0 用于在两个通信应用程序之间提供保密性和数据完整性。

SSL 和 TLS 的区别

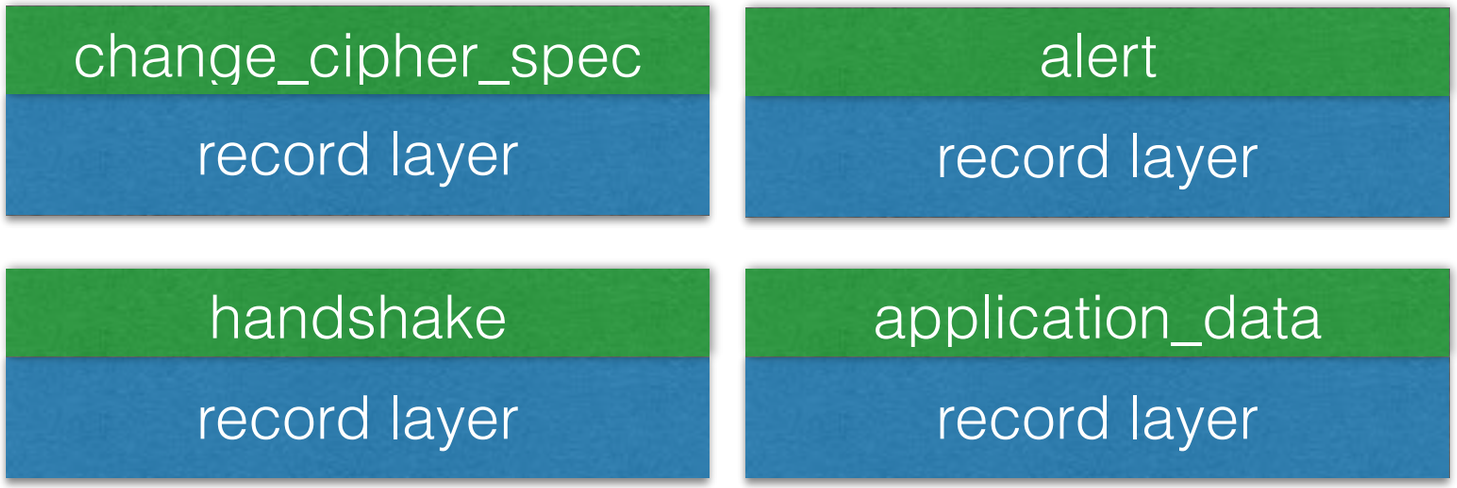
TLS 具有更高的安全性以及适用性

TLS 的产商实现

OpenSSL、Google 的 BoringSSL、Oracle 的 JSSE、Amazon 的 S2n、Microsoft 的 SChannel、Apple 的 Secure Transport

TLS 的具体实现

TLS 是一个分层协议.每一层中的信息可能包含长度、描述和内容等字段。



```
struct {
      uint8 major, minor;
} ProtocolVersion;

enum {
         change_cipher_spec(20), alert(21),        handshake(22),
application_data(23), (255)
} ContentType;

struct {
    ContentType type;
    ProtocolVersion version;
    uint16 length;
    opaque fragment[TLSPlaintext.length];
} TLSPlaintext;
```

ip.src == 180.150.190.136 or ip.dst == 180.150.190.136 | Expression... | + | tcp.port == 80

No.	Time	Source	Destination	Protocol	Length	Info
66	2.660466	172.17.8.1…	180.150.19…	TCP	78	57963 → 443 [SYN] Seq=0 Win=65535 Len=0 MSS.
67	2.692496	180.150.19…	172.17.8.1…	TCP	66	443 → 57963 [SYN, ACK] Seq=0 Ack=1 Win=1460.
68	2.692834	172.17.8.1…	180.150.19…	TCP	54	57963 → 443 [ACK] Seq=1 Ack=1 Win=262144 Le.
69	2.707243	172.17.8.1…	180.150.19…	TLSv1.2	260	Client Hello
70	2.740039	180.150.19…	172.17.8.1…	TCP	54	443 → 57963 [ACK] Seq=1 Ack=207 Win=15872 L.
71	2.740931	180.150.19…	172.17.8.1…	TLSv1.2	1514	Server Hello
72	2.742752	180.150.19…	172.17.8.1…	TCP	1514	[TCP segment of a reassembled PDU]
73	2.742759	180.150.19…	172.17.8.1…	TCP	1514	[TCP segment of a reassembled PDU]
74	2.742865	180.150.19…	172.17.8.1…	TLSv1.2	922	CertificateServer Key Exchange, Server Hell.
75	2.742954	172.17.8.1…	180.150.19…	TCP	54	57963 → 443 [ACK] Seq=207 Ack=2921 Win=2606.

```
▶ Frame 69: 260 bytes on wire (2080 bits), 260 bytes captured (2080 bits) on interface 0
▶ Ethernet II, Src: Apple_cc:45:1a (78:31:c1:cc:45:1a), Dst: Cisco_5c:4a:45 (e8:65:49:5c:4a:45)
▶ Internet Protocol Version 4, Src: 172.17.8.194, Dst: 180.150.190.136
▶ Transmission Control Protocol, Src Port: 57963, Dst Port: 443, Seq: 1, Ack: 1, Len: 206
▼ Secure Sockets Layer
  ▼ TLSv1.2 Record Layer: Handshake Protocol: Client Hello
      Content Type: Handshake (22)
      Version: TLS 1.0 (0x0301)
      Length: 201
    ▶ Handshake Protocol: Client Hello
```

TLS 协商的具体过程

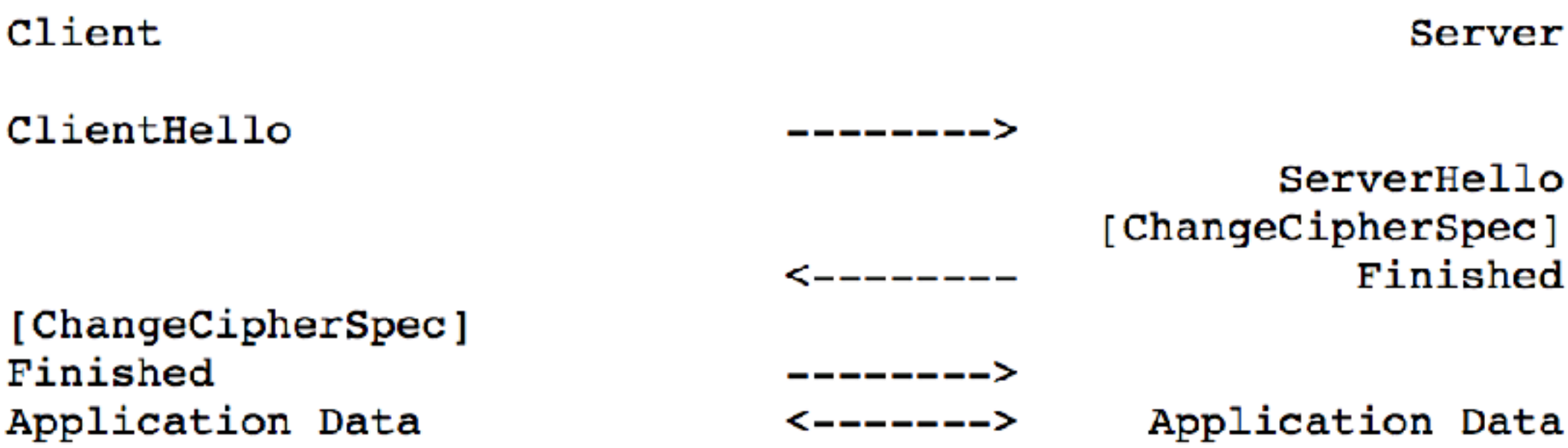
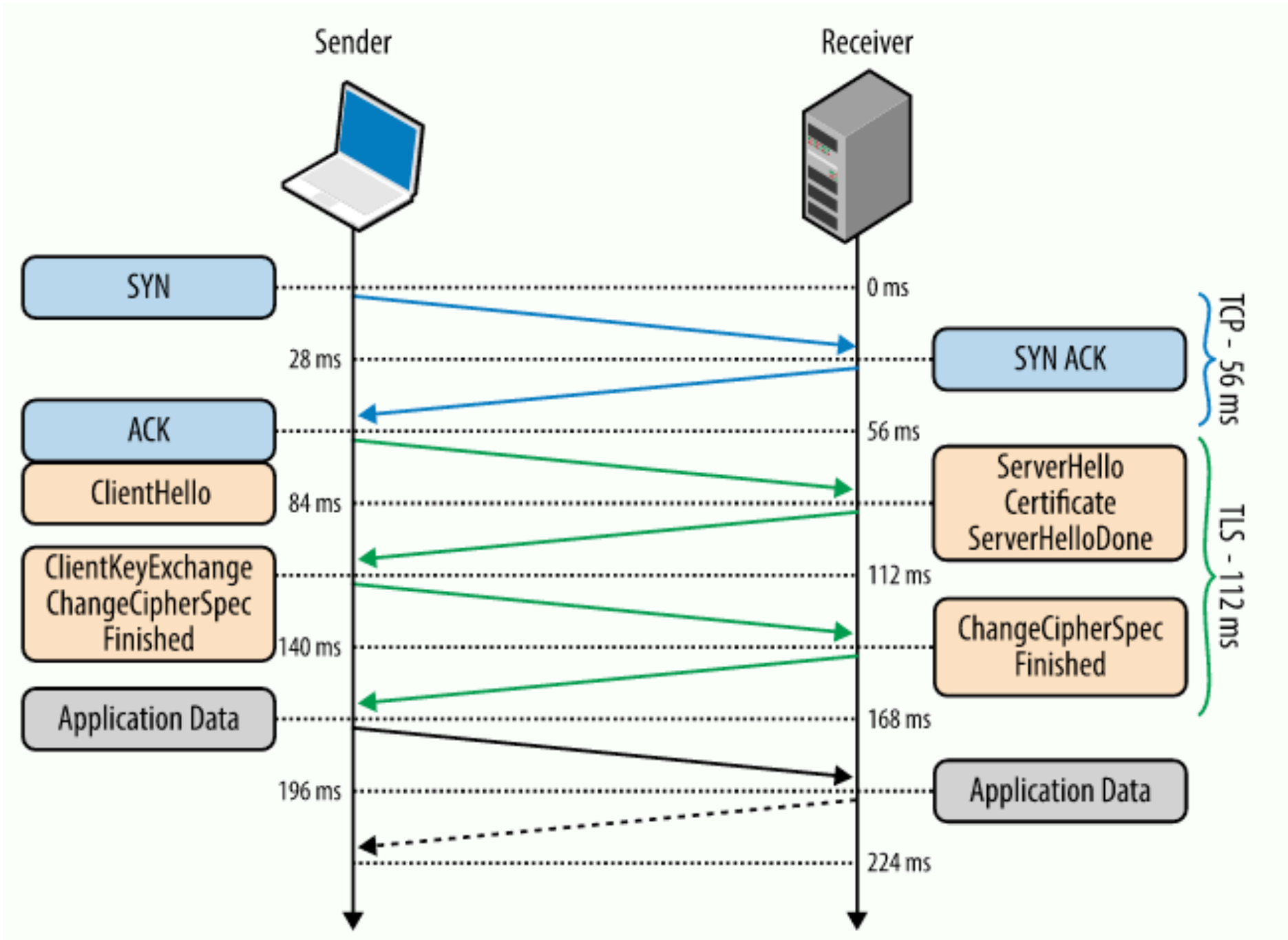


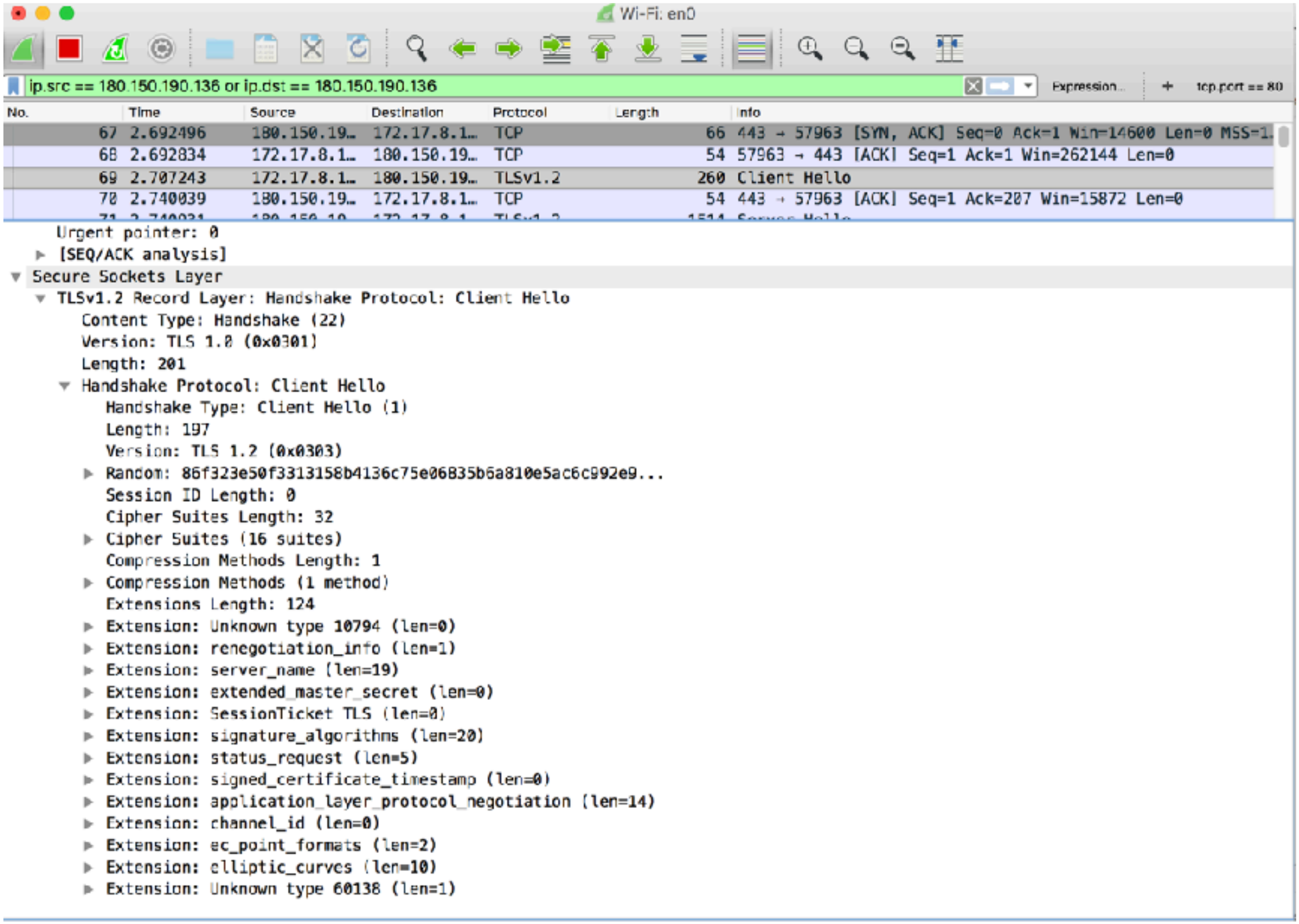
Fig. 2 - Message flow for an abbreviated handshake


Sender
Receiver
SYN
0 ms
28 ms
SYN ACK
TCP - 56 ms
ACK
56 ms
ClientHello
84 ms
ServerHello
Certificate
ServerHelloDone
ClientKeyExchange
ChangeCipherSpec
Finished
112 ms
TLS - 112 ms
140 ms
ChangeCipherSpec
Finished
Application Data
168 ms
196 ms
Application Data
224 ms

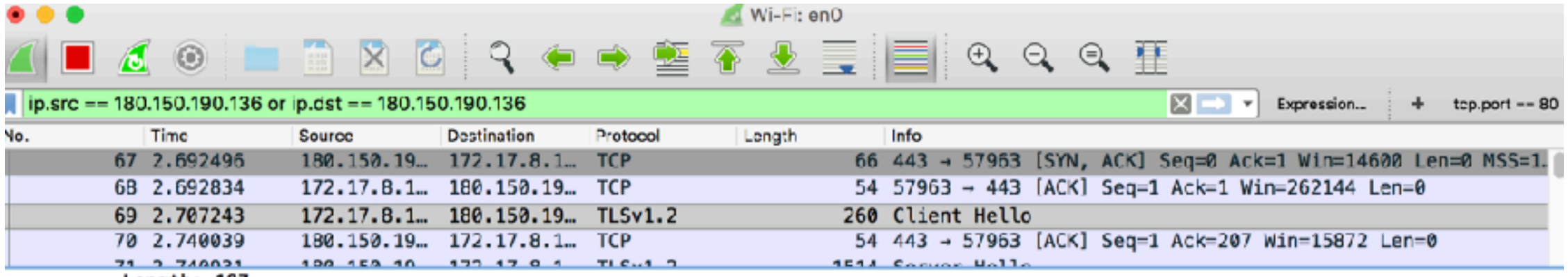
Client Hello

```
struct {
    ProtocolVersion client_version;
    Random random;
    SessionID session_id;
    CipherSuite cipher_suites<2..2^16-2>;
    CompressionMethod compression_methods<1..2^8-1>;
    select (extensions_present) {
        case false:
            struct {};
        case true:
            Extension extensions<0..2^16-1>;
    };
} ClientHello;
```

Client Hello

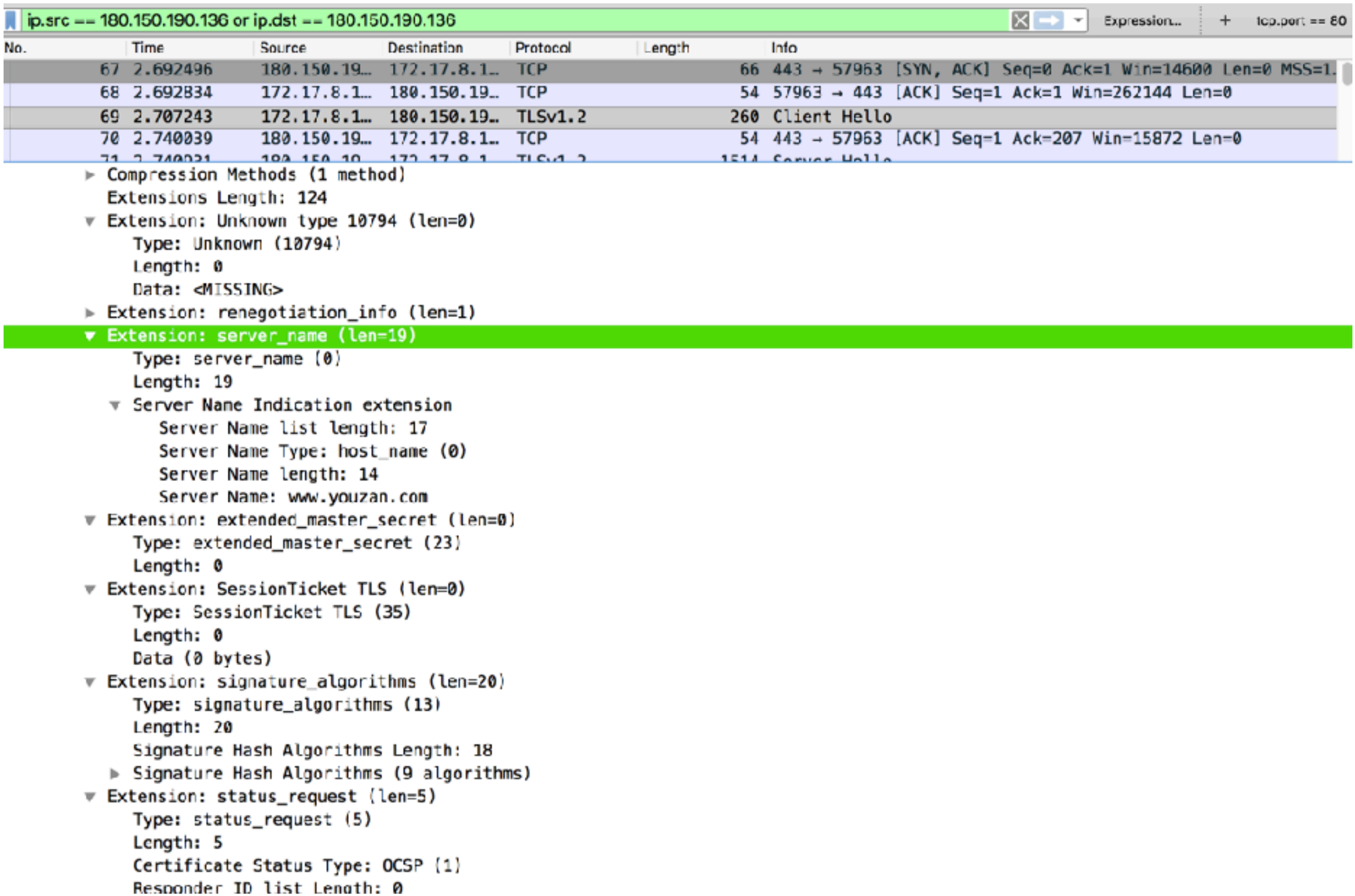


Cipher Suite




```
Length: 197
Version: TLS 1.2 (0x0303)
▶ Random: 86f323e50f3313158b4136c75e06835b6a810e5ac6c992e9...
Session ID Length: 0
Cipher Suites Length: 32
▼ Cipher Suites (16 suites)
    Cipher Suite: Unknown (0x8a8a)
    Cipher Suite: TLS_ECDHE_ECDSA_WITH_AES_128_GCM_SHA256 (0xc02b)
    Cipher Suite: TLS_ECDHE_RSA_WITH_AES_128_GCM_SHA256 (0xc02f)
    Cipher Suite: TLS_ECDHE_ECDSA_WITH_AES_256_GCM_SHA384 (0xc02c)
    Cipher Suite: TLS_ECDHE_RSA_WITH_AES_256_GCM_SHA384 (0xc030)
    Cipher Suite: TLS_ECDHE_ECDSA_WITH_CHACHA20_POLY1305_SHA256 (0xcca9)
    Cipher Suite: TLS_ECDHE_RSA_WITH_CHACHA20_POLY1305_SHA256 (0xcca8)
    Cipher Suite: TLS_ECDHE_ECDSA_WITH_CHACHA20_POLY1305_SHA256 (0xcc14)
    Cipher Suite: TLS_ECDHE_RSA_WITH_CHACHA20_POLY1305_SHA256 (0xcc13)
    Cipher Suite: TLS_ECDHE_RSA_WITH_AES_128_CBC_SHA (0xc013)
    Cipher Suite: TLS_ECDHE_RSA_WITH_AES_256_CBC_SHA (0xc014)
    Cipher Suite: TLS_RSA_WITH_AES_128_GCM_SHA256 (0x009c)
    Cipher Suite: TLS_RSA_WITH_AES_256_GCM_SHA384 (0x009d)
    Cipher Suite: TLS_RSA_WITH_AES_128_CBC_SHA (0x002f)
    Cipher Suite: TLS_RSA_WITH_AES_256_CBC_SHA (0x0035)
    Cipher Suite: TLS_RSA_WITH_3DES_EDE_CBC_SHA (0x000a)
  Compression Methods Length: 1
▶ Compression Methods (1 method)
  Extensions Length: 124
▶ Extension: Unknown type 10794 (len=0)
▶ Extension: renegotiation_info (len=1)
▶ Extension: server_name (len=19)
▶ Extension: extended_master_secret (len=0)
▶ Extension: SessionTicket TLS (len=0)
▶ Extension: signature_algorithms (len=20)
▶ Extension: status_request (len=5)
```

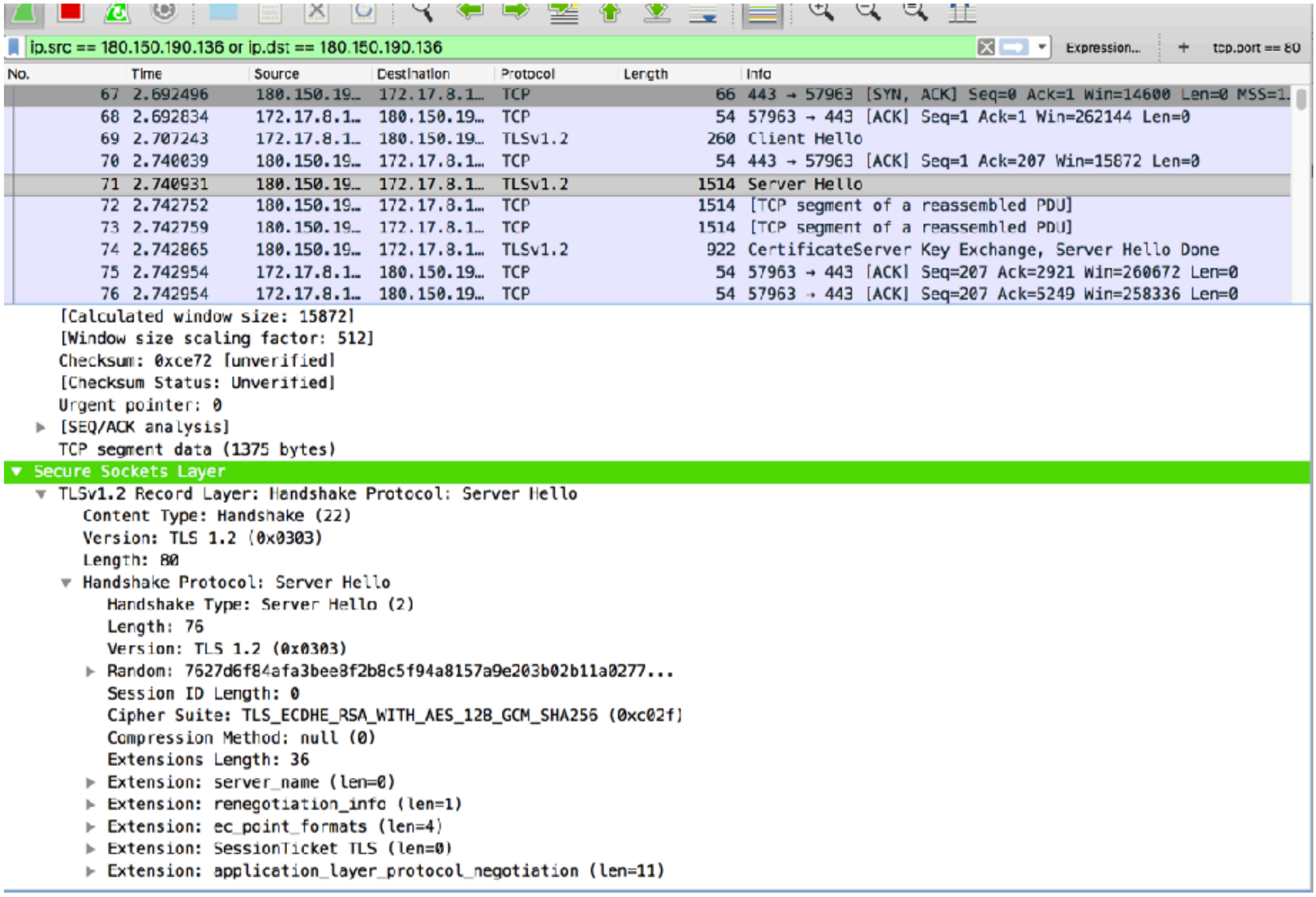
Server Name


ip.src == 180.150.190.136 or ip.dst == 180.150.190.136
Expression...
+
tcp.port == 80
No. Time Source Destination Protocol Length Info
67 2.692496 180.150.19... 172.17.8.1... TCP 66 443 → 57963 [SYN, ACK] Seq=0 Ack=1 Win=14600 Len=0 MSS=1.
68 2.692834 172.17.8.1... 180.150.19... TCP 54 57963 → 443 [ACK] Seq=1 Ack=1 Win=262144 Len=0
69 2.707243 172.17.8.1... 180.150.19... TLSv1.2 260 Client Hello
70 2.740039 180.150.19... 172.17.8.1... TCP 54 443 → 57963 [ACK] Seq=1 Ack=207 Win=15872 Len=0
Compression Methods (1 method)
Extensions Length: 124
Extension: Unknown type 10794 (len=0)
Type: Unknown (10794)
Length: 0
Data: <MISSING>
Extension: renegotiation_info (len=1)
Extension: server_name (len=19)
Type: server_name (0)
Length: 19
Server Name Indication extension
Server Name list length: 17
Server Name Type: host_name (0)
Server Name length: 14
Server Name: www.youzan.com
Extension: extended_master_secret (len=0)
Type: extended_master_secret (23)
Length: 0
Extension: SessionTicket TLS (len=0)
Type: SessionTicket TLS (35)
Length: 0
Data (0 bytes)
Extension: signature_algorithms (len=20)
Type: signature_algorithms (13)
Length: 20
Signature Hash Algorithms Length: 18
Signature Hash Algorithms (9 algorithms)
Extension: status_request (len=5)
Type: status_request (5)
Length: 5
Certificate Status Type: OCSP (1)
Responder ID list Length: 0

Server Hello

```
struct {
    ProtocolVersion server_version;
    Random random;
    SessionID session_id;
    CipherSuite cipher_suite;
    CompressionMethod compression_method;
    select (extensions_present) {
        case false:
            struct {};
        case true:
            Extension extensions<0..2^16-1>;
    };
} ServerHello;
```

Server Hello



Cipher Suite:
TLS_ECDHE_RSA_WITH_AES
_128_GCM_SHA256 (0xc02f)

是什么意思

TLS 协商到底协商什么东西?

最重要的就是协商出cipher suite

一个 cipher suite 是4类算法的组合：

1. key exchange algorithm (密钥交换)
2. authentication method (认证方法)
3. bulk encryption cipher (加密算法)
4. message authentication code (消息认证码)
5. pseudorandom function （伪随机函数）

TLS_ECDHE_RSA_WITH_AES_128_GCM_SHA256

密钥交换: ECDHE

认证方法: RSA

加密算法: AES_128_GCM

MAC 算法：SHA256

```
› /usr/local/openssl-1.1.0c/bin/openssl ciphers -v 'ECDHE-RSA-AES128-GCM-SHA256'

ECDHE-RSA-AES128-GCM-SHA256 TLSv1.2 Kx=ECDH     Au=RSA  Enc=AESGCM(128) Mac=AEAD
```

cipher suite 需要在 IANA 进行注册

https://www.iana.org/assignments/tls-parameters/tls-parameters.xhtml#tls-parameters-4

Server Certificate

Structure of this message:

```
opaque ASN.1Cert<1..2^24-1>;

struct {
    ASN.1Cert certificate_list<0..2^24-1>;
} Certificate;
```

certificate_list
: This is a sequence (chain) of certificates. The sender's certificate MUST come first in the list. Each following certificate MUST directly certify the one preceding it. Because certificate validation requires that root keys be distributed independently, the self-signed certificate that specifies the root certificate authority MAY be omitted from the chain, under the assumption that the remote end must already possess it in order to validate it in any case.

Wi-Fi: en0

ip.src == 180.150.190.136 or ip.dst == 180.150.190.136 | Expression... | + | tcp.port == 80

No.	Time	Source	Destination	Protocol	Length	Info
70	2.740039	180.150.19…	172.17.8.1…	TCP	54	443 → 57963 [ACK] Seq=1 Ack=207 Win=15872 Len=0
71	2.740931	180.150.19…	172.17.8.1…	TLSv1.2	1514	Server Hello
72	2.742752	180.150.19…	172.17.8.1…	TCP	1514	[TCP segment of a reassembled PDU]
73	2.742759	180.150.19…	172.17.8.1…	TCP	1514	[TCP segment of a reassembled PDU]
74	2.742865	180.150.19…	172.17.8.1…	TLSv1.2	922	CertificateServer Key Exchange, Server Hello Done
75	2.742954	172.17.8.1…	180.150.19…	TCP	54	57963 → 443 [ACK] Seq=207 Ack=2921 Win=260672 Len=0
76	2.742954	172.17.8.1…	180.150.19…	TCP	54	57963 → 443 [ACK] Seq=207 Ack=5249 Win=258336 Len=0
77	2.743107	172.17.8.1…	180.150.19…	TCP	54	[TCP Window Update] 57963 → 443 [ACK] Seq=207 Ack=5249 W.
78	2.745934	172.17.8.1…	180.150.19…	TLSv1.2	180	Client Key Exchange, Change Cipher Spec, Hello Request, .
79	2.773229	180.150.19…	172.17.8.1…	TLSv1.2	312	New Session Ticket, Change Cipher Spec, Encrypted Handsh.

```
[Window size scaling factor: 512]
Checksum: 0x39f1 [unverified]
[Checksum Status: Unverified]
Urgent pointer: 0
▶ [SEQ/ACK analysis]
TCP segment data (521 bytes)
▶ [4 Reassembled TCP Segments (4816 bytes): #71(1375), #72(1460), #73(1460), #74(521)]
▼ Secure Sockets Layer
  ▼ TLSv1.2 Record Layer: Handshake Protocol: Certificate
      Content Type: Handshake (22)
      Version: TLS 1.2 (0x0303)
      Length: 4811
    ▼ Handshake Protocol: Certificate
        Handshake Type: Certificate (11)
        Length: 4807
        Certificates Length: 4804
      ▼ Certificates (4804 bytes)
          Certificate Length: 1375
        ▶ Certificate: 3082055b30820443a003020102020900ee92fa98e8f8a5c9... (id-at-commonName=*.youzan.com,id-at-organizationName=\346
          Certificate Length: 1236
        ▶ Certificate: 308204d0308203b8a003020102020107300d06092a864886... (id-at-commonName=Go Daddy Secure Certificate Authority -
          Certificate Length: 1153
        ▶ Certificate: 3082047d30820365a00302010202031be715300d06092a86... (id-at-commonName=Go Daddy Root Certificate Authority - G2
          Certificate Length: 1028
        ▶ Certificate: 30820400308202e8a003020102020100300d06092a864886... (id-at-organizationalUnitName=Go Daddy Class 2 Certificati
▶ Secure Sockets Layer
```

```
› openssl s_client -connect www.youzan.com:443                    10:40
CONNECTED(00000003)
depth=3 /C=US/O=The Go Daddy Group, Inc./OU=Go Daddy Class 2 Certification Authority
verify error:num=19:self signed certificate in certificate chain
verify return:0
---
Certificate chain
 0 s:/OU=Domain Control Validated/CN=*.koudaitong.com
   i:/C=US/ST=Arizona/L=Scottsdale/O=GoDaddy.com, Inc./OU=http://certs.godaddy.com/repository//CN=Go Dadd
y Secure Certificate Authority - G2
 1 s:/C=US/ST=Arizona/L=Scottsdale/O=GoDaddy.com, Inc./OU=http://certs.godaddy.com/repository//CN=Go Dadd
y Secure Certificate Authority - G2
   i:/C=US/ST=Arizona/L=Scottsdale/O=GoDaddy.com, Inc./CN=Go Daddy Root Certificate Authority - G2
 2 s:/C=US/ST=Arizona/L=Scottsdale/O=GoDaddy.com, Inc./CN=Go Daddy Root Certificate Authority - G2
   i:/C=US/O=The Go Daddy Group, Inc./OU=Go Daddy Class 2 Certification Authority
 3 s:/C=US/O=The Go Daddy Group, Inc./OU=Go Daddy Class 2 Certification Authority
   i:/C=US/O=The Go Daddy Group, Inc./OU=Go Daddy Class 2 Certification Authority
```

客户端信任证书

大部分语言使用系统内置的信任证书
Chrome 使用系统内置的信任证书
Firefox 使用 mozilla 维护的信任证书： https://hg.mozilla.org/mozilla-central/raw-file/tip/security/nss/lib/ckfw/builtins/certdata.txt
Linux 使用 /etc/ssl/certs/ca-bundle.trust.crt 同样也来自于 https://hg.mozilla.org/mozilla-central/raw-file/tip/security/nss/lib/ckfw/builtins/certdata.txt

Certificate Request(可选)

要求客户端随后发送证书 (Client Certificate)

ip.src == 192.168.33.10 or ip.dst == 192.168.33.10 | Expression… | + | tcp.port == 80

No.	Time	Source	Destination	Protocol	Length	Info
138	105.433386	192.168.33…	192.168.33…	TCP	66	443 → 50710 [ACK] Seq=1 Ack=224 Win=30080 L.
139	105.435888	192.168.33…	192.168.33…	TLSv1.2	1342	Server Hello, Certificate, Server Key Excha.
140	105.435961	192.168.33…	192.168.33…	TCP	66	50710 → 443 [ACK] Seq=224 Ack=1277 Win=1304.
141	105.829643	192.168.33…	192.168.33…	TLSv1.2	78	Certificate
142	105.829664	192.168.33…	192.168.33…	TLSv1.2	141	Client Key Exchange
143	105.829672	192.168.33…	192.168.33…	TLSv1.2	72	Change Cipher Spec
144	105.829685	192.168.33…	192.168.33…	TLSv1.2	111	Encrypted Handshake Message
145	105.829976	192.168.33…	192.168.33…	TCP	66	443 → 50710 [ACK] Seq=1277 Ack=362 Win=3008.
146	105.830255	192.168.33…	192.168.33…	TLSv1.2	117	Change Cipher Spec, Encrypted Handshake Mes.
147	105.830305	192.168.33…	192.168.33…	TCP	66	50710 → 443 [ACK] Seq=362 Ack=1328 Win=1310.

```
Content Type: Handshake (22)
Version: TLS 1.2 (0x0303)
Length: 63
▼ Handshake Protocol: Certificate Request
    Handshake Type: Certificate Request (13)
    Length: 55
    Certificate types count: 3
  ▶ Certificate types (3 types)
    Signature Hash Algorithms Length: 30
  ▶ Signature Hash Algorithms (15 algorithms)
    Distinguished Names Length: 17
  ▶ Distinguished Names (17 bytes)
▼ Handshake Protocol: Server Hello Done
    Handshake Type: Server Hello Done (14)
    Length: 0
```

Server Key Exchange Message

This message will be sent immediately after the server Certificate message (or the ServerHello message, if this is an anonymous negotiation).

The ServerKeyExchange message is sent by the server only when the server Certificate message (if sent) does not contain enough data to allow the client to exchange a premaster secret. This is true for the following key exchange methods:

DHE_DSS
DHE_RSA
DH_anon

这条消息只会在 Server Certificate 无法记录足够的数据去交换 premaster secret时，才会被发送

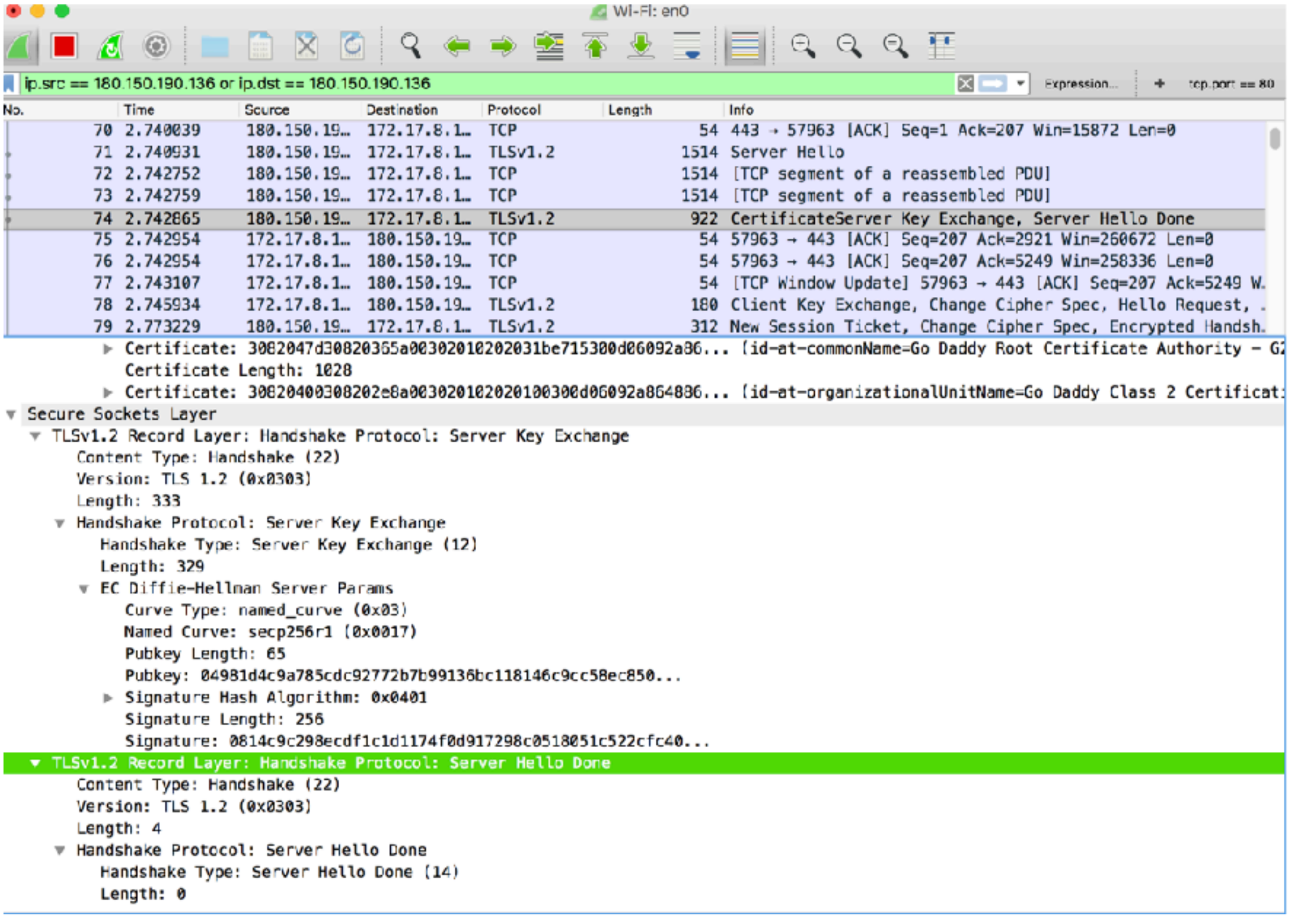
Wi-Fi: en0

ip.src == 180.150.190.136 or ip.dst == 180.150.190.136 | Expression... | + | tcp.port == 80

No.	Time	Source	Destination	Prctocol	Length	Info
70	2.740039	180.150.19…	172.17.8.1…	TCP	54	443 → 57963 [ACK] Seq=1 Ack=207 Win=15872 Len=0
71	2.740931	180.150.19…	172.17.8.1…	TLSv1.2	1514	Server Hello
72	2.742752	180.150.19…	172.17.8.1…	TCP	1514	[TCP segment of a reassembled PDU]
73	2.742759	180.150.19…	172.17.8.1…	TCP	1514	[TCP segment of a reassembled PDU]
74	2.742865	180.150.19…	172.17.8.1…	TLSv1.2	922	CertificateServer Key Exchange, Server Hello Done
75	2.742954	172.17.8.1…	180.150.19…	TCP	54	57963 → 443 [ACK] Seq=207 Ack=2921 Win=260672 Len=0
76	2.742954	172.17.8.1…	180.150.19…	TCP	54	57963 → 443 [ACK] Seq=207 Ack=5249 Win=258336 Len=0
77	2.743107	172.17.8.1…	180.150.19…	TCP	54	[TCP Window Update] 57963 → 443 [ACK] Seq=207 Ack=5249 W.
78	2.745934	172.17.8.1…	180.150.19…	TLSv1.2	180	Client Key Exchange, Change Cipher Spec, Hello Request, .
79	2.773229	180.150.19…	172.17.8.1…	TLSv1.2	312	New Session Ticket, Change Cipher Spec, Encrypted Handsh.

```
    [Window size scaling factor: 512]
    Checksum: 0x39f1 [unverified]
    [Checksum Status: Unverified]
    Urgent pointer: 0
  ▶ [SEQ/ACK analysis]
    TCP segment data (521 bytes)
▶ [4 Reassembled TCP Segments (4816 bytes): #71(1375), #72(1460), #73(1460), #74(521)]
▼ Secure Sockets Layer
  ▶ TLSv1.2 Record Layer: Handshake Protocol: Certificate
▼ Secure Sockets Layer
  ▼ TLSv1.2 Record Layer: Handshake Protocol: Server Key Exchange
      Content Type: Handshake (22)
      Version: TLS 1.2 (0x0303)
      Length: 333
    ▼ Handshake Protocol: Server Key Exchange
        Handshake Type: Server Key Exchange (12)
        Length: 329
      ▼ EC Diffie-Hellman Server Params
          Curve Type: named_curve (0x03)
          Named Curve: secp256r1 (0x0017)
          Pubkey Length: 65
          Pubkey: 04981d4c9a785cdc92772b7b99136bc118146c9cc58ec850...
        ▶ Signature Hash Algorithm: 0x0401
          Signature Length: 256
          Signature: 0814c9c298ecdf1c1d1174f0d917298c0518051c522cfc40...
▶ TLSv1.2 Record Layer: Handshake Protocol: Server Hello Done
```

Server Hello Done



Client Certificate(可选)

ip.src == 192.168.33.10 or ip.dst == 192.168.33.10 | Expression... | + | tcp.port == 80

No.	Time	Source	Destination	Protocol	Length	Info
181	233.355635	192.168.33…	192.168.33…	TLSv1.2	826	Certificate
182	233.355666	192.168.33…	192.168.33…	TLSv1.2	141	Client Key Exchange
183	233.355681	192.168.33…	192.168.33…	TLSv1.2	335	Certificate Verify
184	233.355701	192.168.33…	192.168.33…	TLSv1.2	72	Change Cipher Spec
185	233.355722	192.168.33…	192.168.33…	TLSv1.2	111	Encrypted Handshake Message
186	233.355942	192.168.33…	192.168.33…	TCP	66	443 → 50725 [ACK] Seq=1277 Ack=1353 Win=316.
187	233.356179	192.168.33…	192.168.33…	TCP	66	443 → 50725 [ACK] Seq=1277 Ack=1673 Win=331.
188	233.356987	192.168.33…	192.168.33…	TLSv1.2	117	Change Cipher Spec, Encrypted Handshake Mes.
189	233.357050	192.168.33…	192.168.33…	TCP	66	50725 → 443 [ACK] Seq=1673 Ack=1328 Win=131.
190	233.358104	192.168.33…	192.168.33…	TLSv1.2	452	Application Data

```
▶ Ethernet II, Src: 0a:00:27:00:00:02 (0a:00:27:00:00:02), Dst: PcsCompu_72:af:e8 (08:00:27:72:af:e8)
▶ Internet Protocol Version 4, Src: 192.168.33.1, Dst: 192.168.33.10
▶ Transmission Control Protocol, Src Port: 50725, Dst Port: 443, Seq: 518, Ack: 1277, Len: 760
▼ Secure Sockets Layer
  ▼ TLSv1.2 Record Layer: Handshake Protocol: Certificate
      Content Type: Handshake (22)
      Version: TLS 1.2 (0x0303)
      Length: 755
    ▼ Handshake Protocol: Certificate
        Handshake Type: Certificate (11)
        Length: 751
        Certificates Length: 748
      ▼ Certificates (748 bytes)
          Certificate Length: 745
        ▶ Certificate: 308202e5308201cd020101300d06092a864886f70d01010b... (id-at-commonName=client,id-at-organizationN
```

Client Key Exchange Message

ip.src == 192.168.33.10 or ip.dst == 192.168.33.10 Expression... + tcp.port == 80

No.	Time	Source	Destination	Protocol	Length	Info
181	233.355635	192.168.33…	192.168.33…	TLSv1.2	826	Certificate
182	233.355666	192.168.33…	192.168.33…	TLSv1.2	141	Client Key Exchange
183	233.355681	192.168.33…	192.168.33…	TLSv1.2	335	Certificate Verify
184	233.355701	192.168.33…	192.168.33…	TLSv1.2	72	Change Cipher Spec
185	233.355722	192.168.33…	192.168.33…	TLSv1.2	111	Encrypted Handshake Message
186	233.355942	192.168.33…	192.168.33…	TCP	66	443 → 50725 [ACK] Seq=1277 Ack=1353 Win=316
187	233.356179	192.168.33…	192.168.33…	TCP	66	443 → 50725 [ACK] Seq=1277 Ack=1673 Win=331
188	233.356987	192.168.33…	192.168.33…	TLSv1.2	117	Change Cipher Spec, Encrypted Handshake Mes
189	233.357050	192.168.33…	192.168.33…	TCP	66	50725 → 443 [ACK] Seq=1673 Ack=1328 Win=131
190	233.358104	192.168.33…	192.168.33…	TLSv1.2	452	Application Data

```
▶ Ethernet II, Src: 0a:00:27:00:00:02 (0a:00:27:00:00:02), Dst: PcsCompu_72:af:e8 (08:00:27:72:af:e8)
▶ Internet Protocol Version 4, Src: 192.168.33.1, Dst: 192.168.33.10
▶ Transmission Control Protocol, Src Port: 50725, Dst Port: 443, Seq: 518, Ack: 1277, Len: 760
▼ Secure Sockets Layer
  ▼ TLSv1.2 Record Layer: Handshake Protocol: Certificate
      Content Type: Handshake (22)
      Version: TLS 1.2 (0x0303)
      Length: 755
    ▼ Handshake Protocol: Certificate
        Handshake Type: Certificate (11)
        Length: 751
        Certificates Length: 748
      ▼ Certificates (748 bytes)
          Certificate Length: 745
        ▶ Certificate: 308202e5308201cd020101300d06092a864886f70d01010b... (id-at-commonName=client,id-at-organizationN
```

Encrypted Handshake Message

Capturing from vboxnet2

ip.src == 192.168.33.10 or ip.dst == 192.168.33.10 — Expression… — tcp.port == 80

No.	Time	Source	Destination	Protocol	Length	Info
186	233.355942	192.168.33…	192.168.33…	TCP	66	443 → 50725 [ACK] Seq=1277 Ack=1353 Win=316.
187	233.356179	192.168.33…	192.168.33…	TCP	66	443 → 50725 [ACK] Seq=1277 Ack=1673 Win=331.
188	233.356987	192.168.33…	192.168.33…	TLSv1.2	117	Change Cipher Spec, Encrypted Handshake Mes.
189	233.357050	192.168.33…	192.168.33…	TCP	66	50725 → 443 [ACK] Seq=1673 Ack=1328 Win=131.
190	233.358104	192.168.33…	192.168.33…	TLSv1.2	452	Application Data
191	233.358501	192.168.33…	192.168.33…	TLSv1.2	490	Application Data
192	233.358555	192.168.33…	192.168.33…	TCP	66	50725 → 443 [ACK] Seq=2059 Ack=1752 Win=130.
193	233.358757	192.168.33…	192.168.33…	TCP	66	50725 → 443 [FIN, ACK] Seq=2059 Ack=1752 Wi.
194	233.359047	192.168.33…	192.168.33…	TLSv1.2	97	Encrypted Alert
195	233.359051	192.168.33…	192.168.33…	TCP	66	443 → 50725 [FIN, ACK] Seq=1783 Ack=2059 Wi.

- Frame 188: 117 bytes on wire (936 bits), 117 bytes captured (936 bits) on interface 0
- Ethernet II, Src: PcsCompu_72:af:e8 (08:00:27:72:af:e8), Dst: 0a:00:27:00:00:02 (0a:00:27:00:00:02)
- Internet Protocol Version 4, Src: 192.168.33.10, Dst: 192.168.33.1
- Transmission Control Protocol, Src Port: 443, Dst Port: 50725, Seq: 1277, Ack: 1673, Len: 51
- Secure Sockets Layer
  - TLSv1.2 Record Layer: Change Cipher Spec Protocol: Change Cipher Spec
    - Content Type: Change Cipher Spec (20)
    - Version: TLS 1.2 (0x0303)
    - Length: 1
    - Change Cipher Spec Message
  - **TLSv1.2 Record Layer: Handshake Protocol: Encrypted Handshake Message**
    - Content Type: Handshake (22)
    - Version: TLS 1.2 (0x0303)
    - Length: 40
    - Handshake Protocol: Encrypted Handshake Message

Change Cipher Spec Protocol

The change cipher spec protocol exists to signal transitions in ciphering strategies. The protocol consists of a single message, which is encrypted and compressed under the current (not the pending) connection state. The message consists of a single byte of value 1.

```
struct {
    enum { change_cipher_spec(1), (255) } type;
} ChangeCipherSpec;
```

The ChangeCipherSpec message is sent by both the client and the server to notify the receiving party that subsequent records will be protected under the newly negotiated CipherSpec and keys. Reception of this message causes the receiver to instruct the record layer to immediately copy the read pending state into the read current state. Immediately after sending this message, the sender MUST instruct the record layer to make the write pending state the write active state.

通知对方接下来的数据将是加密的

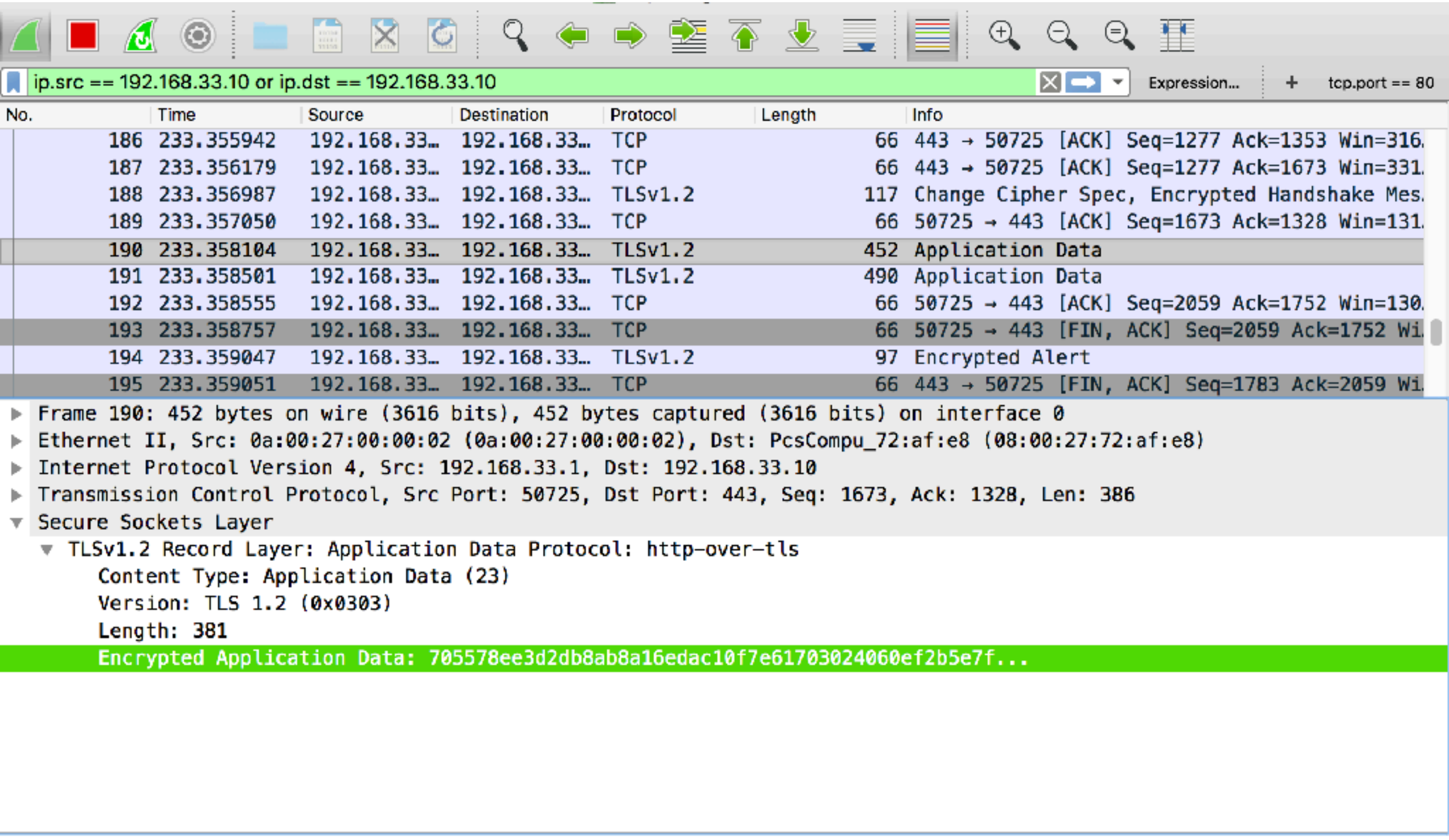
Capturing from vboxnet2

ip.src == 192.168.33.10 or ip.dst == 192.168.33.10 | Expression... | + | tcp.port == 80

No.	Time	Source	Destination	Protocol	Length	Info
186	233.355942	192.168.33…	192.168.33…	TCP	66	443 → 50725 [ACK] Seq=1277 Ack=1353 Win=316
187	233.356179	192.168.33…	192.168.33…	TCP	66	443 → 50725 [ACK] Seq=1277 Ack=1673 Win=331
188	233.356987	192.168.33…	192.168.33…	TLSv1.2	117	Change Cipher Spec, Encrypted Handshake Mes
189	233.357050	192.168.33…	192.168.33…	TCP	66	50725 → 443 [ACK] Seq=1673 Ack=1328 Win=131
190	233.358104	192.168.33…	192.168.33…	TLSv1.2	452	Application Data
191	233.358501	192.168.33…	192.168.33…	TLSv1.2	490	Application Data
192	233.358555	192.168.33…	192.168.33…	TCP	66	50725 → 443 [ACK] Seq=2059 Ack=1752 Win=130
193	233.358757	192.168.33…	192.168.33…	TCP	66	50725 → 443 [FIN, ACK] Seq=2059 Ack=1752 Wi
194	233.359047	192.168.33…	192.168.33…	TLSv1.2	97	Encrypted Alert
195	233.359051	192.168.33…	192.168.33…	TCP	66	443 → 50725 [FIN, ACK] Seq=1783 Ack=2059 Wi

```
▶ Frame 188: 117 bytes on wire (936 bits), 117 bytes captured (936 bits) on interface 0
▶ Ethernet II, Src: PcsCompu_72:af:e8 (08:00:27:72:af:e8), Dst: 0a:00:27:00:00:02 (0a:00:27:00:00:02)
▶ Internet Protocol Version 4, Src: 192.168.33.10, Dst: 192.168.33.1
▶ Transmission Control Protocol, Src Port: 443, Dst Port: 50725, Seq: 1277, Ack: 1673, Len: 51
▼ Secure Sockets Layer
  ▼ TLSv1.2 Record Layer: Change Cipher Spec Protocol: Change Cipher Spec
      Content Type: Change Cipher Spec (20)
      Version: TLS 1.2 (0x0303)
      Length: 1
      Change Cipher Spec Message
  ▼ TLSv1.2 Record Layer: Handshake Protocol: Encrypted Handshake Message
      Content Type: Handshake (22)
      Version: TLS 1.2 (0x0303)
      Length: 40
      Handshake Protocol: Encrypted Handshake Message
```

Application Data



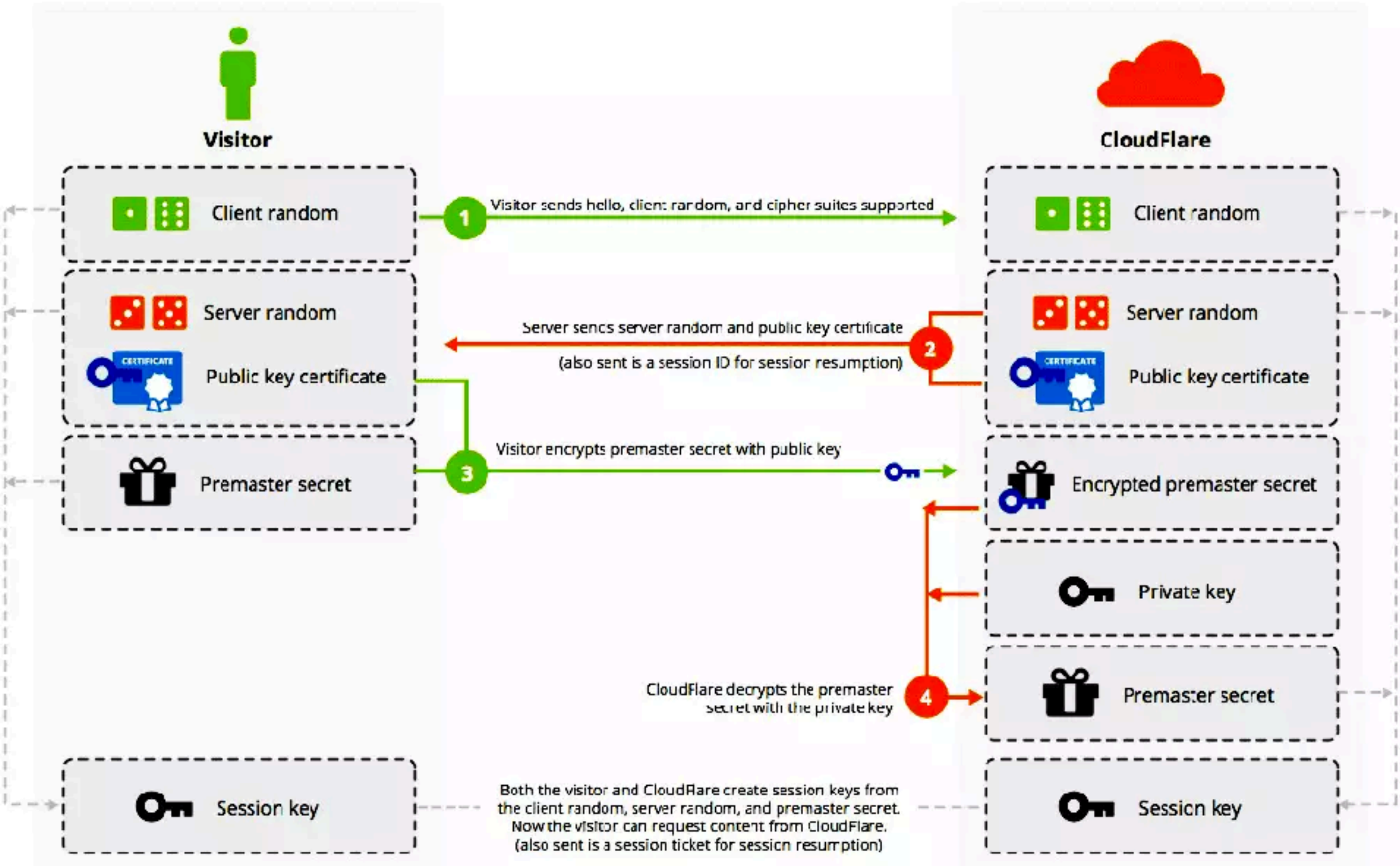
TLS 如何查看 Application Data

重点在于算出 master key

master_secret = PRF(pre_master_secret, "master secret",
ClientHello.random + ServerHello.random)

ClientHello.random 和 ServerHello.random 是明文，只有pre_master_secret不知道

RSA 是密钥协商算法


Visitor
CloudFlare
Client random
1
Visitor sends hello, client random, and cipher suites supported
Client random
Server random
Public key certificate
Server sends server random and public key certificate
(also sent is a session ID for session resumption)
2
Server random
Public key certificate
Premaster secret
3
Visitor encrypts premaster secret with public key
Encrypted premaster secret
Private key
CloudFlare decrypts the premaster secret with the private key
4
Premaster secret
Session key
Both the visitor and CloudFlare create session keys from the client random, server random, and premaster secret. Now the visitor can request content from CloudFlare. (also sent is a session ticket for session resumption)
Session key

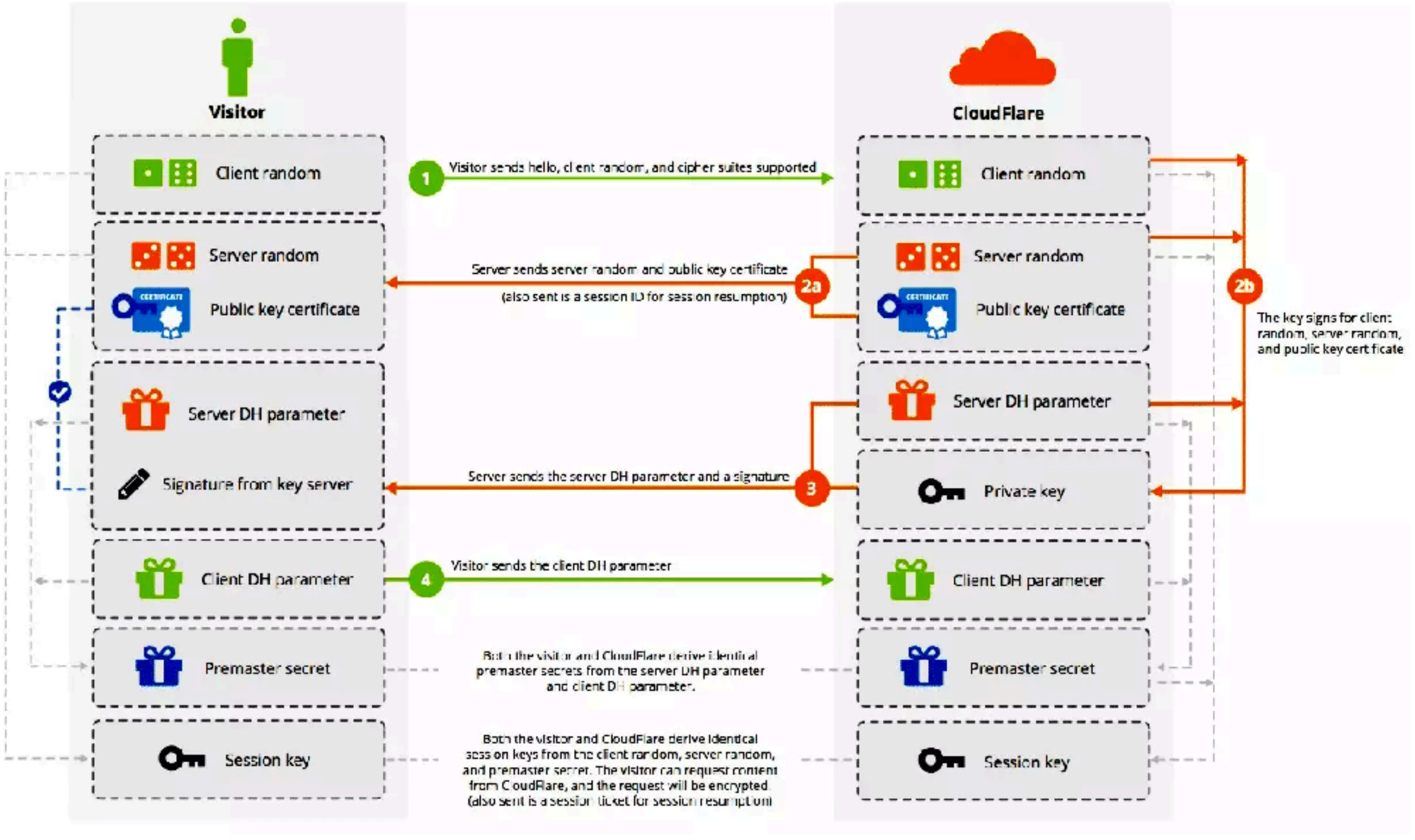
客户端用服务端的证书公钥加密 pre_master_secret, 服务端获取之后，用私约解密即可获得 pre_master_secret.

所以只要获得服务端证书私钥就能正确解密 TLS 流量

RSA 作为密钥协商的最大问题

pre master key 是静态的，不具有所谓的前向安全性(Forward Secrecy), HTTP2 协议要求 SSL 的密钥协商算法需要具有 Perfect Forward Secrecy

DH 是密钥协商算法


Visitor
CloudFlare
Client random
Visitor sends hello, client random, and cipher suites supported
1
Client random
Server random
Server sends server random and public key certificate
(also sent is a session ID for session resumption)
2a
Server random
Public key certificate
Public key certificate
2b
The key signs for client random, server random, and public key certificate
Server DH parameter
Server DH parameter
Signature from key server
Server sends the server DH parameter and a signature
3
Private key
Client DH parameter
Visitor sends the client DH parameter
4
Client DH parameter
Premaster secret
Both the visitor and CloudFlare derive identical premaster secrets from the server DH parameter and client DH parameter.
Premaster secret
Session key
Both the visitor and CloudFlare derive identical session keys from the client random, server random, and premaster secret. The visitor can request content from CloudFlare, and the request will be encrypted.
(also sent is a session ticket for session resumption)
Session key

不再直接传递 pre master key, 而是传递 DH 算法所需要的参数，双方各自生成一对公钥和私钥，并把各自的公钥交给对方，DH 算法保证双方能在不同的参数下生成同样的 pre master key
因为不知道各自的私钥，所以中间人无法获取 pre master key。

浏览器的 SSLKEYLOGFILE

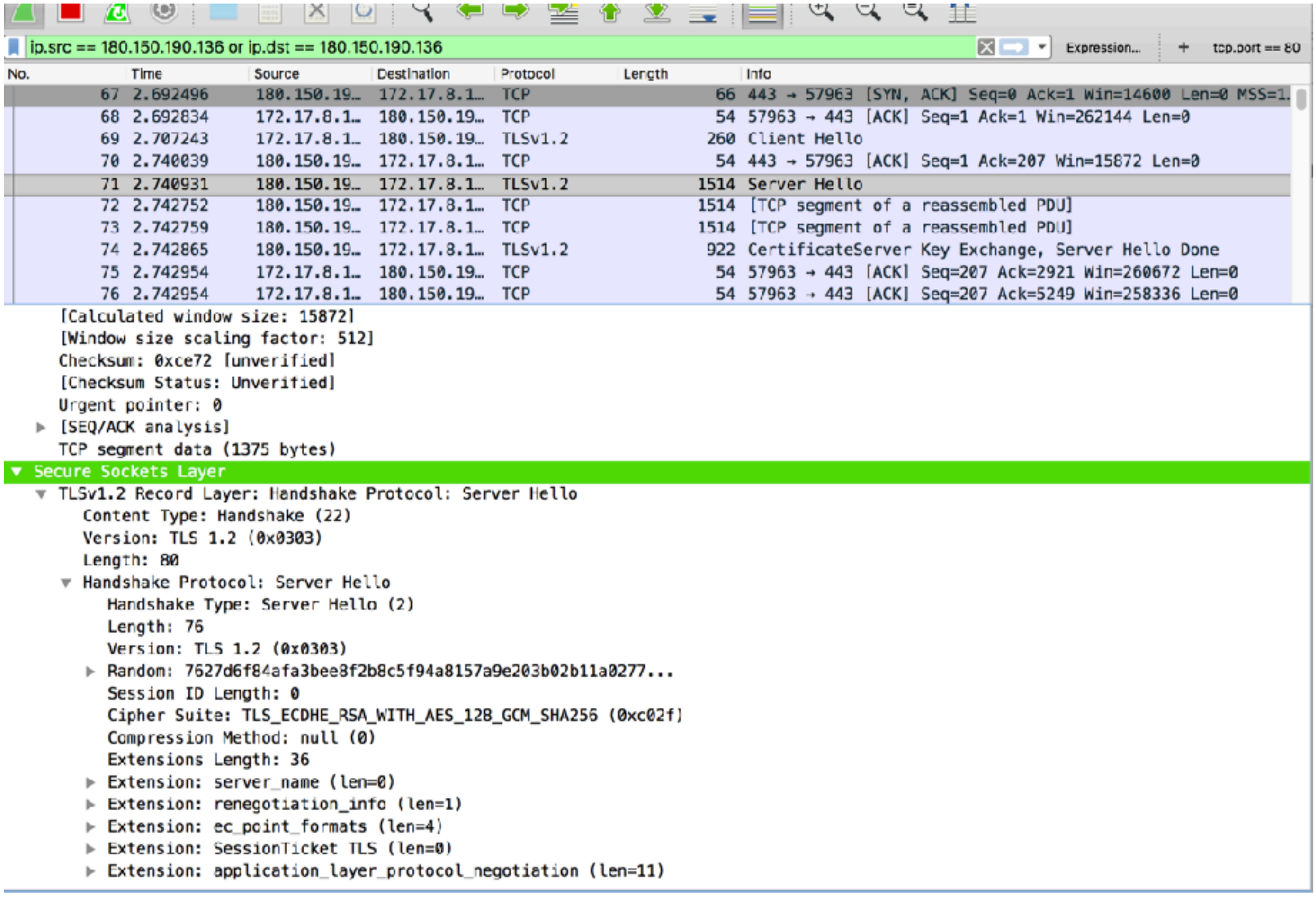
浏览器会把每次 TLS 协商的 master key 放到这个文件，因此配合 wireshark，就能看到 TLS 加密的数据

加速 TLS 协商

几个维度

1. 减少传输的负荷
2. 使用更快的算法
3. 缓存协商的结果
4. TLS 协议上优化

减少传输的负荷


ip.src == 180.150.190.136 or ip.dst == 180.150.190.136
Expression...
tcp.port == 80
No. Time Source Destination Protocol Length Info
67 2.692496 180.150.19… 172.17.8.1… TCP 66 443 → 57963 [SYN, ACK] Seq=0 Ack=1 Win=14600 Len=0 MSS=1.
68 2.692834 172.17.8.1… 180.150.19… TCP 54 57963 → 443 [ACK] Seq=1 Ack=1 Win=262144 Len=0
69 2.707243 172.17.8.1… 180.150.19… TLSv1.2 260 Client Hello
70 2.740039 180.150.19… 172.17.8.1… TCP 54 443 → 57963 [ACK] Seq=1 Ack=207 Win=15872 Len=0
71 2.740931 180.150.19… 172.17.8.1… TLSv1.2 1514 Server Hello
72 2.742752 180.150.19… 172.17.8.1… TCP 1514 [TCP segment of a reassembled PDU]
73 2.742759 180.150.19… 172.17.8.1… TCP 1514 [TCP segment of a reassembled PDU]
74 2.742865 180.150.19… 172.17.8.1… TLSv1.2 922 CertificateServer Key Exchange, Server Hello Done
75 2.742954 172.17.8.1… 180.150.19… TCP 54 57963 → 443 [ACK] Seq=207 Ack=2921 Win=260672 Len=0
76 2.742954 172.17.8.1… 180.150.19… TCP 54 57963 → 443 [ACK] Seq=207 Ack=5249 Win=258336 Len=0
[Calculated window size: 15872]
[Window size scaling factor: 512]
Checksum: 0xce72 [unverified]
[Checksum Status: Unverified]
Urgent pointer: 0
[SEQ/ACK analysis]
TCP segment data (1375 bytes)
Secure Sockets Layer
TLSv1.2 Record Layer: Handshake Protocol: Server Hello
Content Type: Handshake (22)
Version: TLS 1.2 (0x0303)
Length: 80
Handshake Protocol: Server Hello
Handshake Type: Server Hello (2)
Length: 76
Version: TLS 1.2 (0x0303)
Random: 7627d6f84afa3bee8f2b8c5f94a8157a9e203b02b11a0277...
Session ID Length: 0
Cipher Suite: TLS_ECDHE_RSA_WITH_AES_128_GCM_SHA256 (0xc02f)
Compression Method: null (0)
Extensions Length: 36
Extension: server_name (len=0)
Extension: renegotiation_info (len=1)
Extension: ec_point_formats (len=4)
Extension: SessionTicket TLS (len=0)
Extension: application_layer_protocol_negotiation (len=11)

减少传输的证书

1. 只配置自己的证书和中间证书

2. 使用 ECC 证书

可以选择 ECC（Elliptic Curve Cryptography，椭圆曲线密码学）证书。256 位的 ECC Key 等同于 3072 位的 RSA Key.

对称加密 Key 长度	RSA Key 长度	ECC Key 长度
80	1024	160
112	2048	224
128	3072	256
192	7680	384
256	15360	521

ECC 证书的缺点

1. 服务商需要支持生成 ECC 的证书 （可以解决）
2. 兼容性较差，需要较高的系统 （可以解决）

使用更快的算法

在 ARM 平台上使用谷歌的 ChaCha20-Poly1305 算法， 算法针对 ARM 做了优化，在移动设备上使用速度更快、更省电

在 intel 平台上使用优先使用 AES-GCM， Intel 提供了 AES NI（Advanced Encryption Standard new instructions）的 x86 指令集扩展，从硬件上提供对 AES 的支持

AESDEC—Perform One Round of an AES Decryption Flow

Opcode/ Instruction	Op/ En	64/32-bit Mode	CPUID Feature Flag	Description
66 0F 38 DE /r AESDEC xmm1, xmm2/m128	RM	V/V	AES	Perform one round of an AES decryption flow, using the Equivalent Inverse Cipher, operating on a 128-bit data (state) from xmm1 with a 128-bit round key from xmm2/m128.
VEX.NDS.128.66.0F38.WIG DE /r VAESDEC xmm1, xmm2, xmm3/m128	RVM	V/V	Both AES and AVX flags	Perform one round of an AES decryption flow, using the Equivalent Inverse Cipher, operating on a 128-bit data (state) from xmm2 with a 128-bit round key from xmm3/m128; store the result in xmm1.

缓存协商的结果

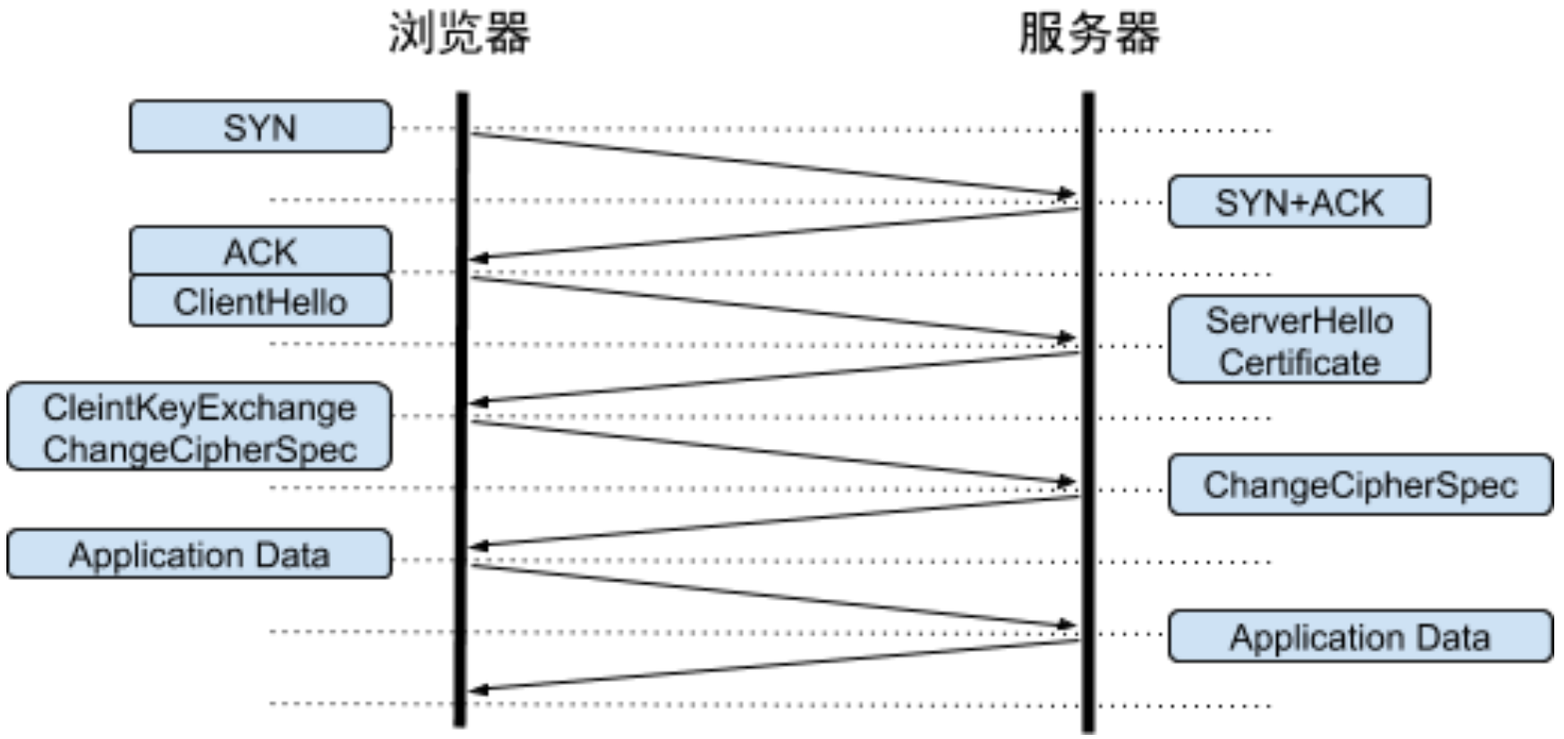
1. 使用 TLS session id 缓存结果

缺点：服务端各个节点需同步 session id

2. 使用 TLS session ticket 缓存结果

缺点：需要轮转 session ticket，避免 session ticket 前向安全性

TLS 协议上优化

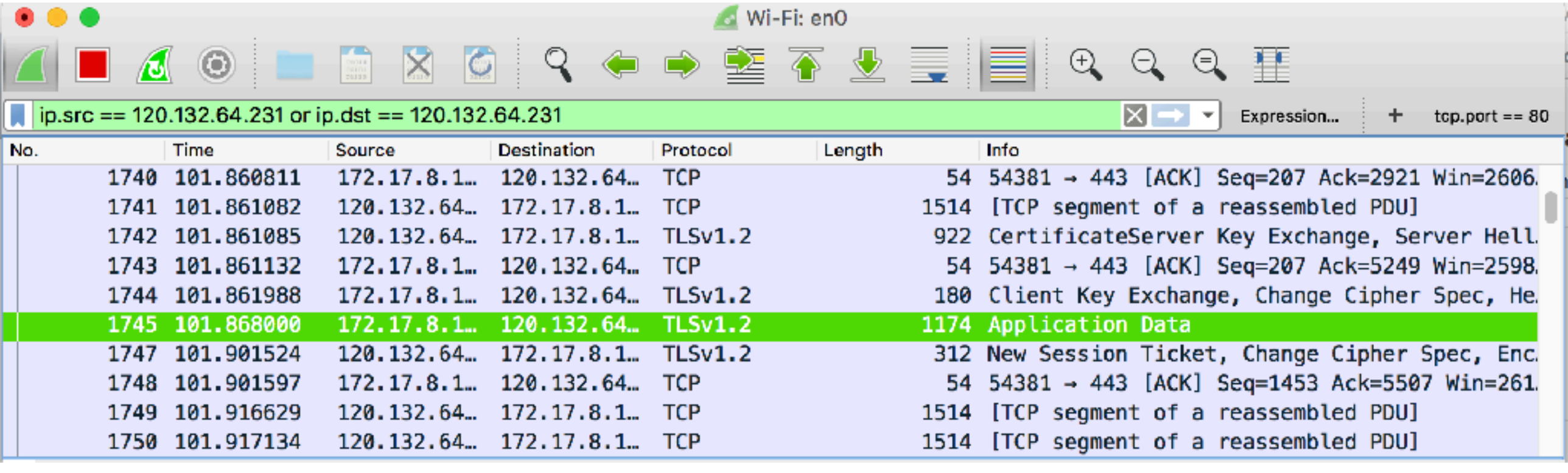


需要两个 RTT 来协商

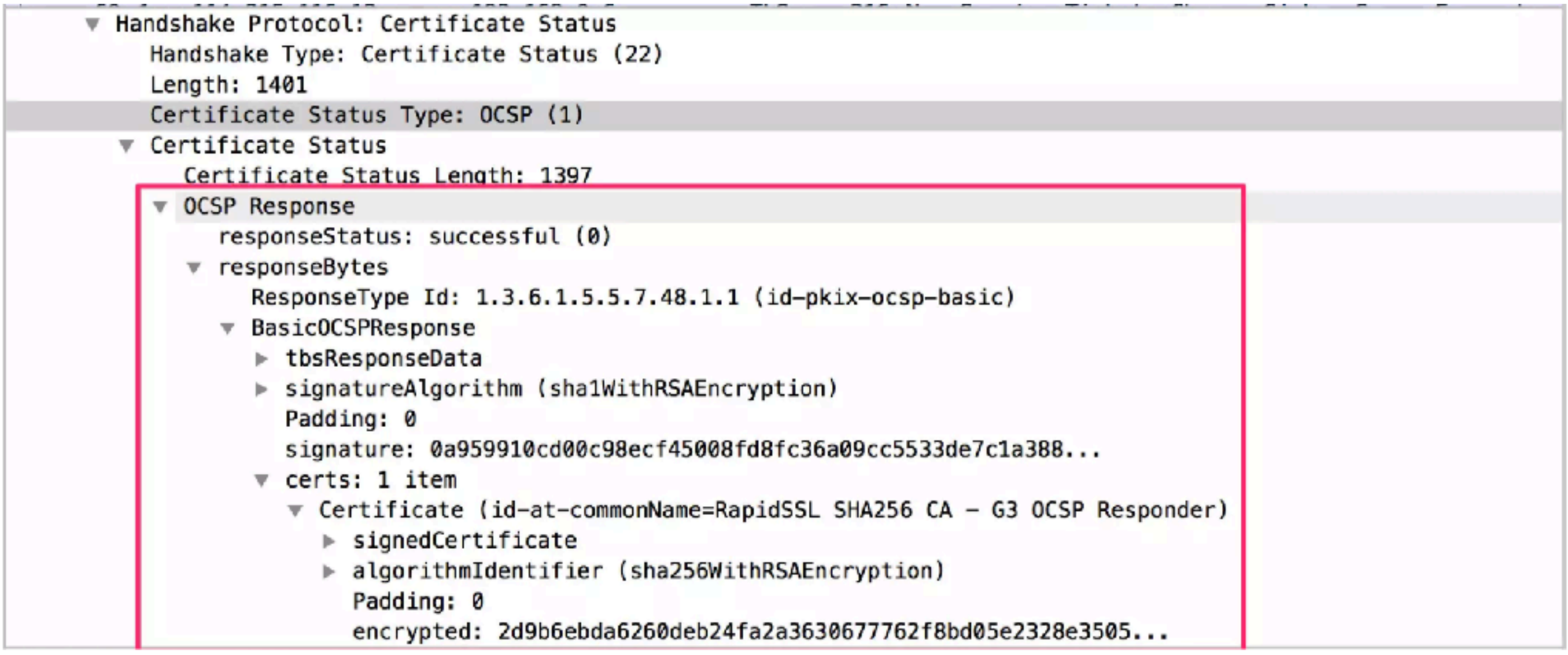
开启 False Start（抢跑）

开启TLS False Start需要浏览器端和服务器端同时满足条件。

Chrome和Firefox需要支持NPN/ALPN，并且服务器端的 cipher suite 支持前向安全（Forward Secrecy）；



使用 OCSP Stapling 代替 OCSP 和 CRL



使用 TLS 1.3

目前只有 Chrome 和 Firefox 默认开启了 TLS 1.3