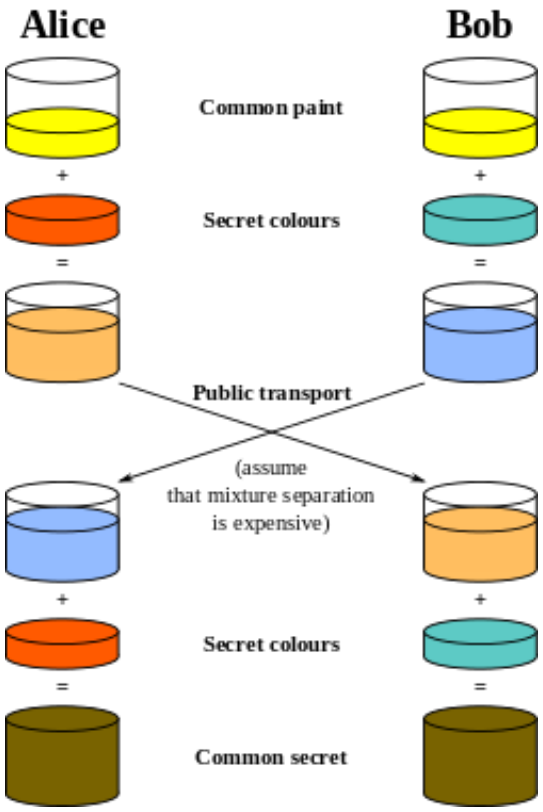
Diffie- Hellman 密钥交换算法

DH算法属于公钥加密算法

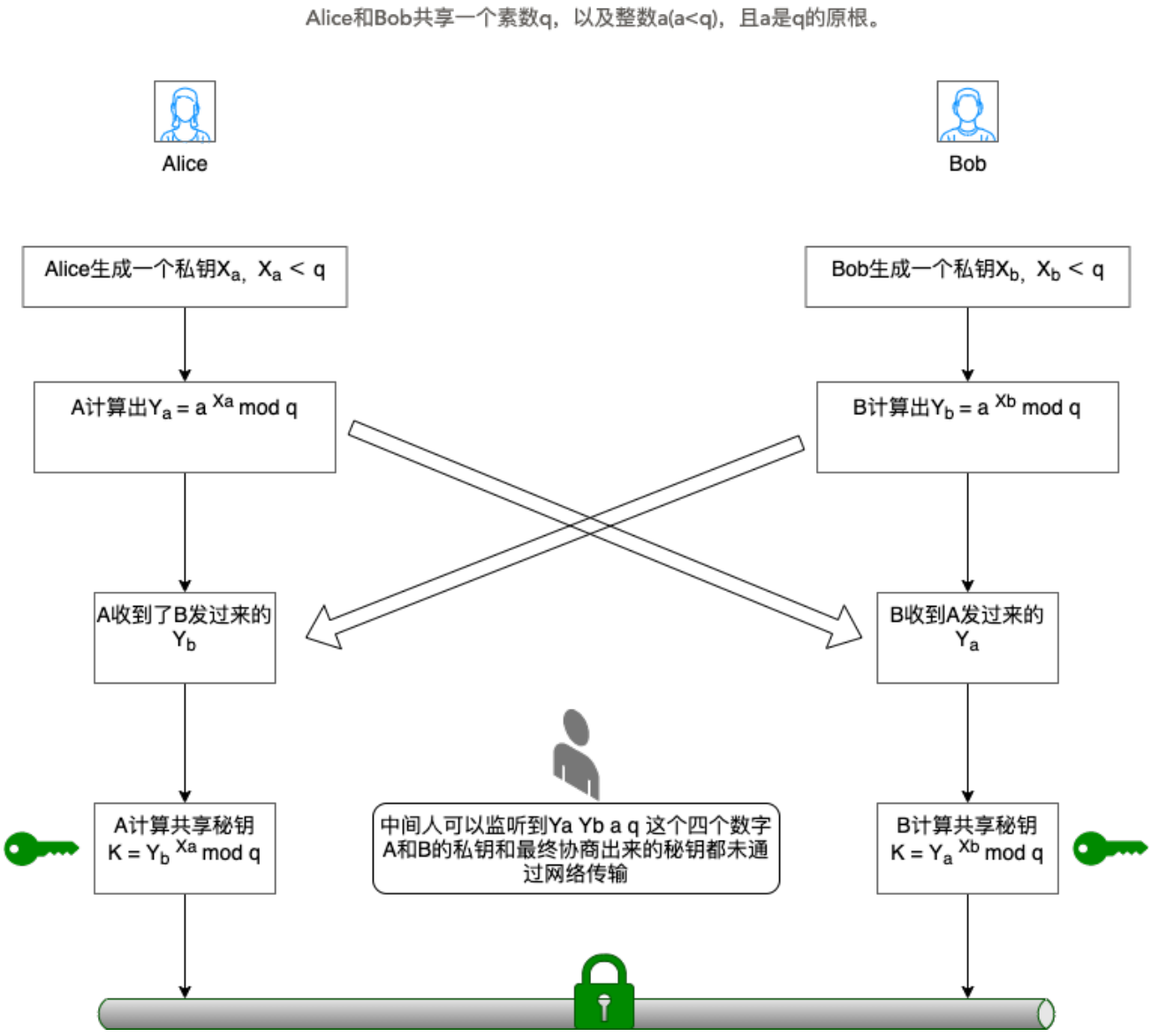
公钥加密算法加解密复杂，花费时间久

加解密数据时使用**对称密码算法**，**密钥管理**使用**公钥密码**技术

其解决问题的主要思想可以用下图来解释



具体交换的过程如下：



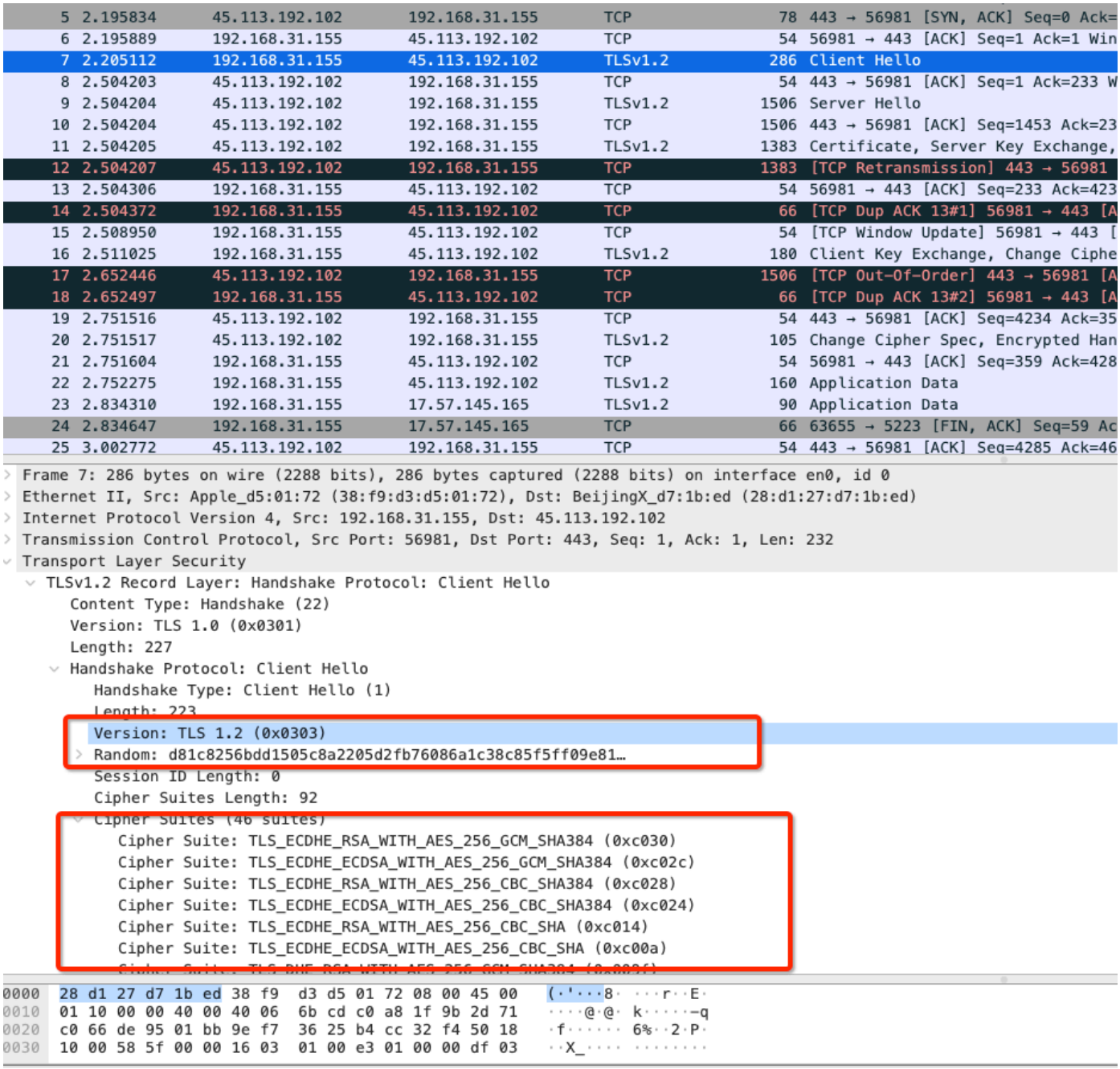
1. Alice和Bob协商一个有限循环群G和它的一个生成元g，一个大素数p;
2. Alice生成一个随机数a，计算$A = g^a mod\ p$，将A发送给Bob；
3. Bob生成一个随机数b，计算$B = g^b mod\ p$，将B发送给Alice；
4. Alice计算$K = B^a mod\ p = (g^b)^a mod\ p$，得到共享密钥$K$；
5. Bob计算$K = A^b\ mod\ p = (g^a)^b\ mod\ p$，得到共享密钥$K$；

> $(g^b)^a = (g^a)^b$因为群是乘法交换的，涉及到数论及代数的内容。Alice和Bob同时协商出K，作为共享密钥。

TLS握手过程

1.Client Hello

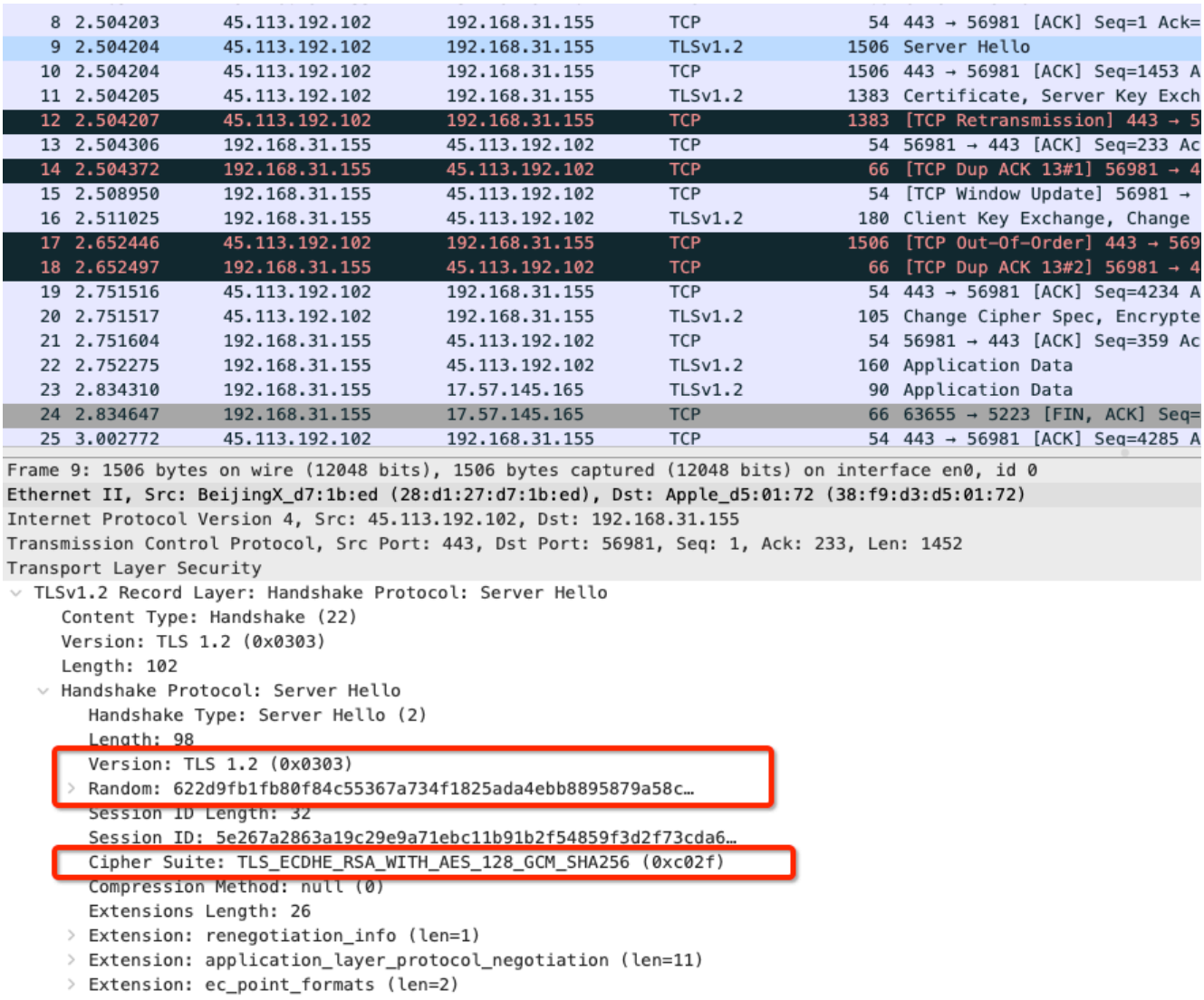
- TLS Version，TLS的版本号
- Cipher Suite，客户端支持的加密套件列表；加密算法秘钥长度等等
    - Client Random，随机数，



2. Server Hello

- TLS Version
- 服务端选中的加密组件，从客户端发送过来的加密套件中选择其中一个

- Server Random，服务端生成的随机数

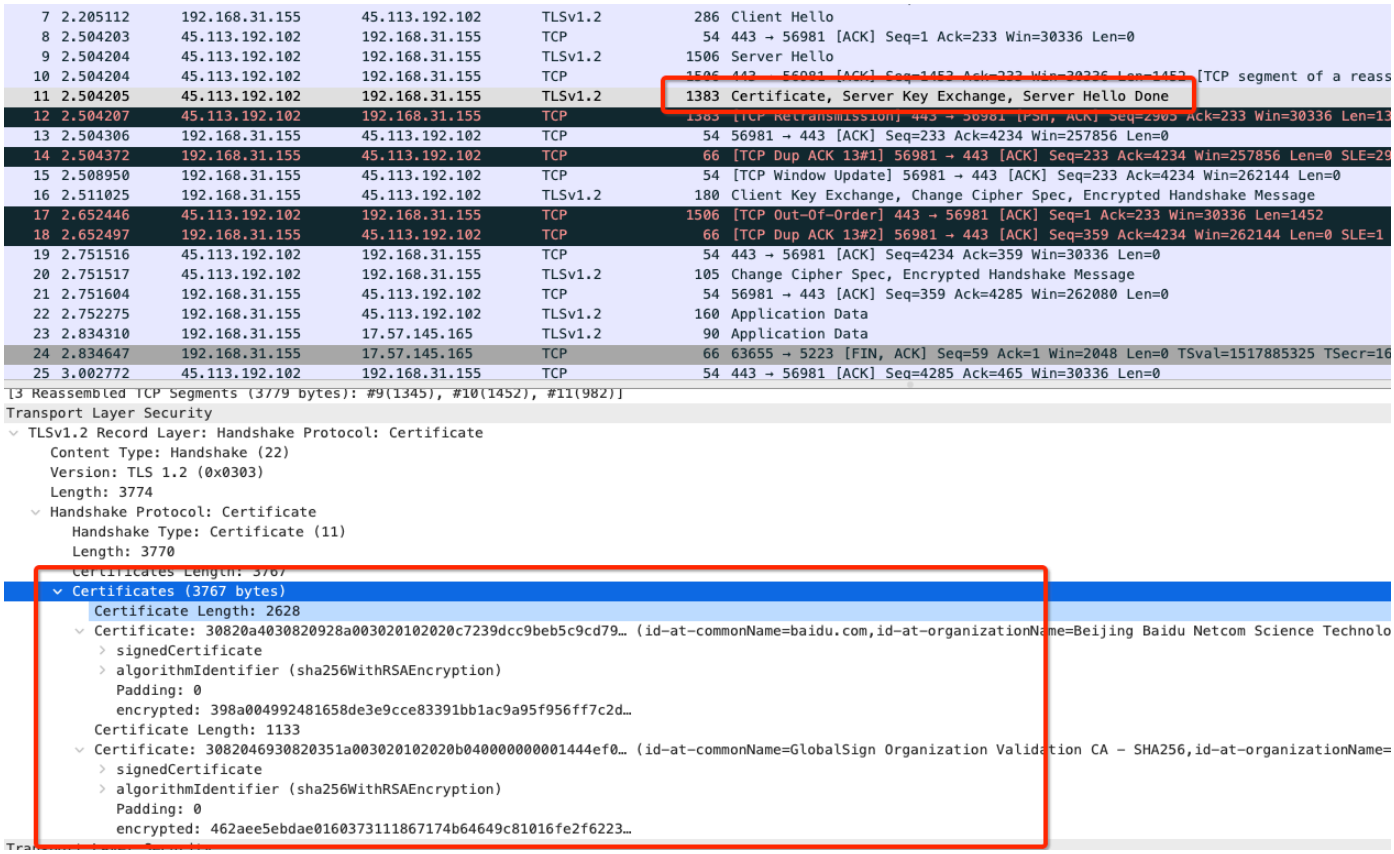


```
    8 2.504203    45.113.192.102  192.168.31.155  TCP       54 443 → 56981 [ACK] Seq=1 Ack=
    9 2.504204    45.113.192.102  192.168.31.155  TLSv1.2 1506 Server Hello
   10 2.504204    45.113.192.102  192.168.31.155  TCP     1506 443 → 56981 [ACK] Seq=1453 A
   11 2.504205    45.113.192.102  192.168.31.155  TLSv1.2 1383 Certificate, Server Key Exch
   12 2.504207    45.113.192.102  192.168.31.155  TCP     1383 [TCP Retransmission] 443 → 5
   13 2.504306    192.168.31.155  45.113.192.102  TCP       54 56981 → 443 [ACK] Seq=233 Ac
   14 2.504372    192.168.31.155  45.113.192.102  TCP       66 [TCP Dup ACK 13#1] 56981 → 4
   15 2.508950    192.168.31.155  45.113.192.102  TCP       54 [TCP Window Update] 56981 →
   16 2.511025    192.168.31.155  45.113.192.102  TLSv1.2 180 Client Key Exchange, Change
   17 2.652446    45.113.192.102  192.168.31.155  TCP     1506 [TCP Out-Of-Order] 443 → 569
   18 2.652497    192.168.31.155  45.113.192.102  TCP       66 [TCP Dup ACK 13#2] 56981 → 4
   19 2.751516    45.113.192.102  192.168.31.155  TCP       54 443 → 56981 [ACK] Seq=4234 A
   20 2.751517    45.113.192.102  192.168.31.155  TLSv1.2 105 Change Cipher Spec, Encrypte
   21 2.751604    192.168.31.155  45.113.192.102  TCP       54 56981 → 443 [ACK] Seq=359 Ac
   22 2.752275    192.168.31.155  45.113.192.102  TLSv1.2 160 Application Data
   23 2.834310    192.168.31.155  17.57.145.165   TLSv1.2   90 Application Data
   24 2.834647    192.168.31.155  17.57.145.165   TCP       66 63655 → 5223 [FIN, ACK] Seq=
   25 3.002772    45.113.192.102  192.168.31.155  TCP       54 443 → 56981 [ACK] Seq=4285 A

Frame 9: 1506 bytes on wire (12048 bits), 1506 bytes captured (12048 bits) on interface en0, id 0
Ethernet II, Src: BeijingX_d7:1b:ed (28:d1:27:d7:1b:ed), Dst: Apple_d5:01:72 (38:f9:d3:d5:01:72)
Internet Protocol Version 4, Src: 45.113.192.102, Dst: 192.168.31.155
Transmission Control Protocol, Src Port: 443, Dst Port: 56981, Seq: 1, Ack: 233, Len: 1452
Transport Layer Security
▾ TLSv1.2 Record Layer: Handshake Protocol: Server Hello
    Content Type: Handshake (22)
    Version: TLS 1.2 (0x0303)
    Length: 102
  ▾ Handshake Protocol: Server Hello
      Handshake Type: Server Hello (2)
      Length: 98
      Version: TLS 1.2 (0x0303)
    › Random: 622d9fb1fb80f84c55367a734f1825ada4ebb8895879a58c…
      Session ID Length: 32
      Session ID: 5e267a2863a19c29e9a71ebc11b91b2f54859f3d2f73cda6…
      Cipher Suite: TLS_ECDHE_RSA_WITH_AES_128_GCM_SHA256 (0xc02f)
      Compression Method: null (0)
      Extensions Length: 26
    › Extension: renegotiation_info (len=1)
    › Extension: application_layer_protocol_negotiation (len=11)
    › Extension: ec_point_formats (len=2)
```

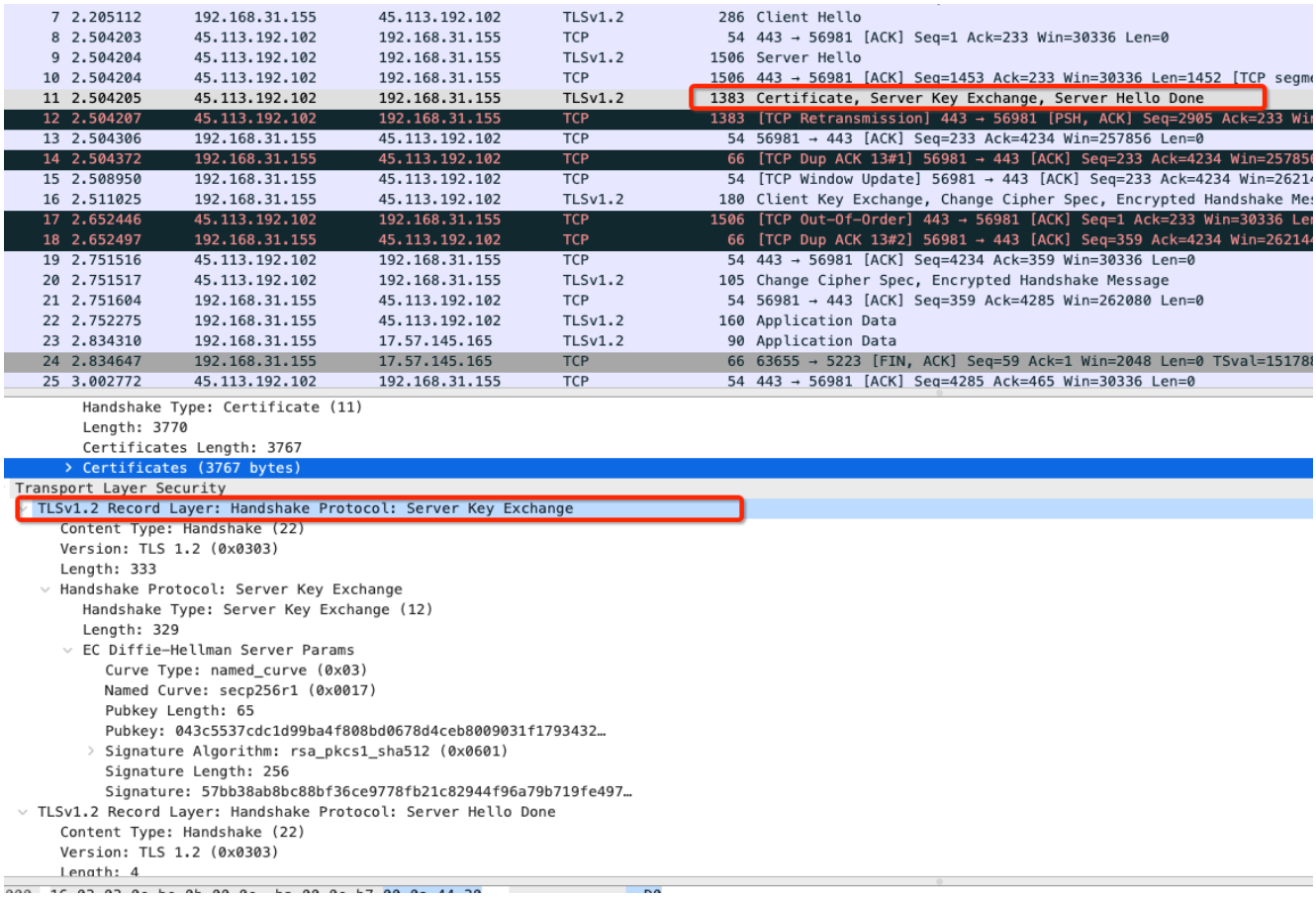
3. Certificate

- 服务器的公钥证书 (被CA签过名的)



4. Server Key Exchange

- Server Params，ECDHE算法需要用到



> ECDHE是一种秘钥交换算法
>
> 为了防止伪造，Server Params是经过服务器的私钥签名的

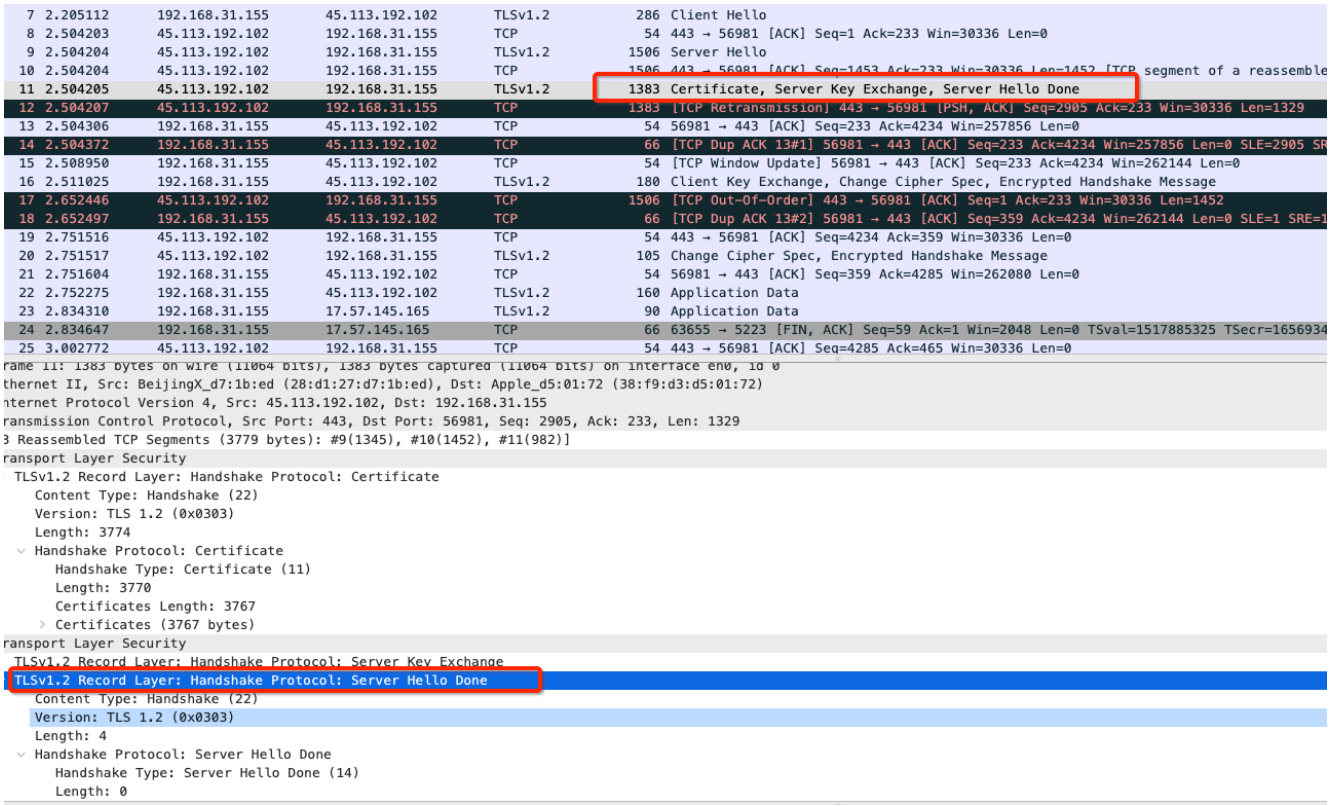
5. Server Hello Done

   告诉客户端：协商部分结束

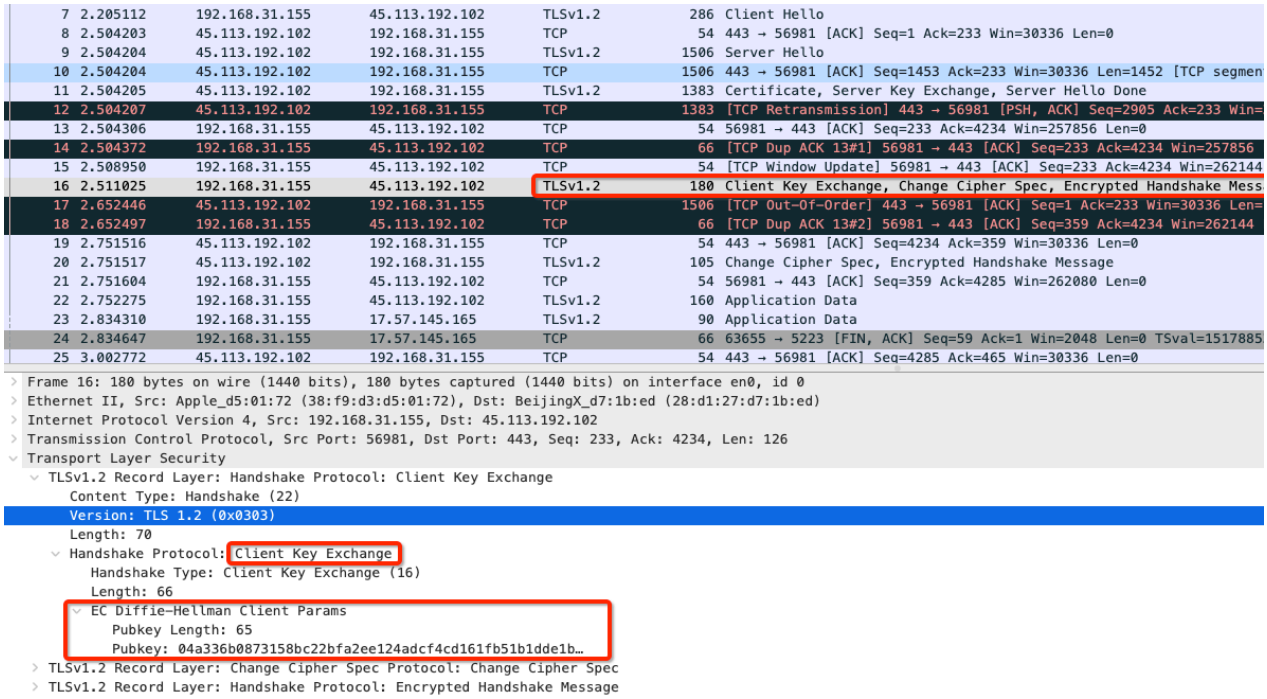
   目前为止，客户端和服务器端共享了

```
1. Client Random
2. Server Random
3. Server Params
```

   并且，客户端已经得到了服务器给过来的证书（包含公钥），接下来客户端需要验证证书是否真实有效



6. Client Key Exchange
   - Client Params，ECDHE算法需要用到

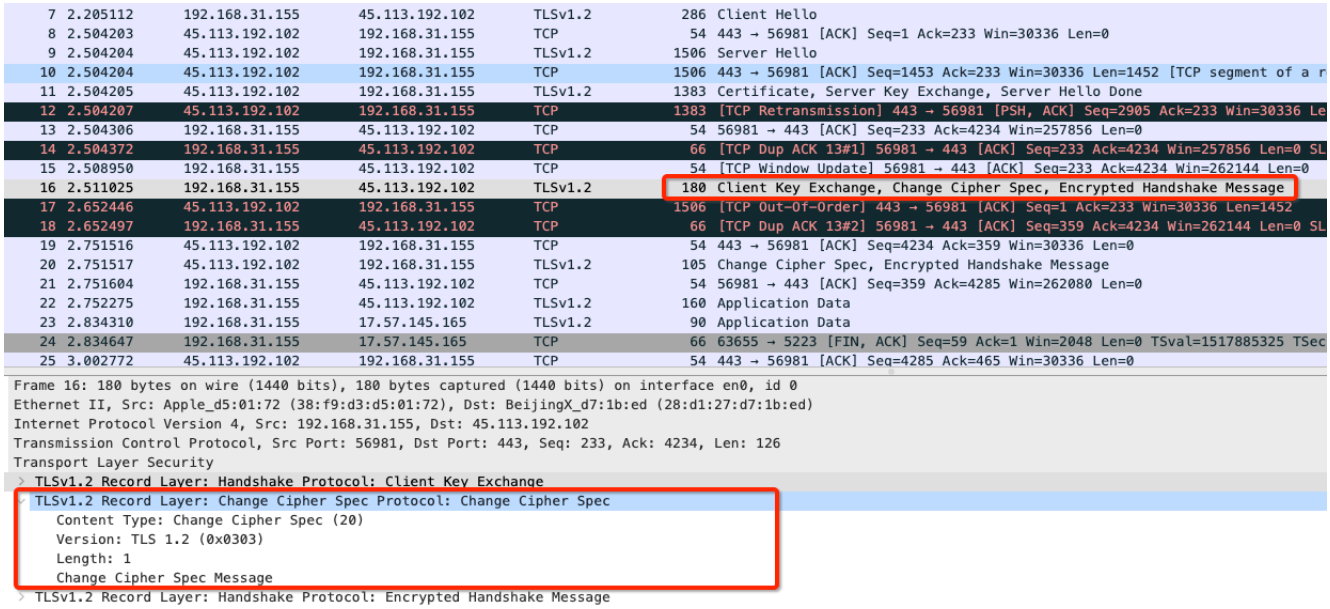


到这一步，客户端和服务端都拥有了Server Params 、Client Params；

客户端和服务器端都可以使用ECDHE算法根据Server Params、Client Params计算出一个新的随机秘钥串Pre-master Secret

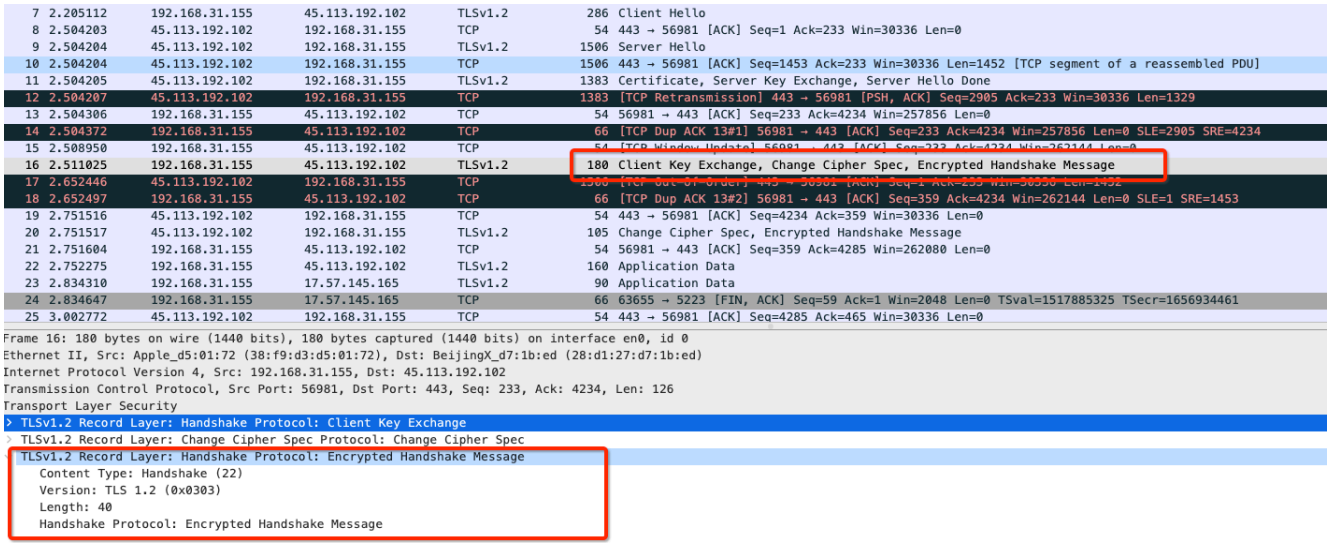
7. Change Cipher Spec

   告知服务器之后的通信会采用计算出来的会话秘钥进行加密



8. Finished

   1. 包括链接到这一步所有的报文的摘要值，加密之后发送给服务器
   2. 这次握手是否成功要以服务器是否能够正确解密报文作为判断标准



9. Change Cipher Spec
10. Finished
    1. 到此为止，客户端服务器都验证了加解密没有问题，握手正式结束
    2. 后面开始传输加密的请求和响应