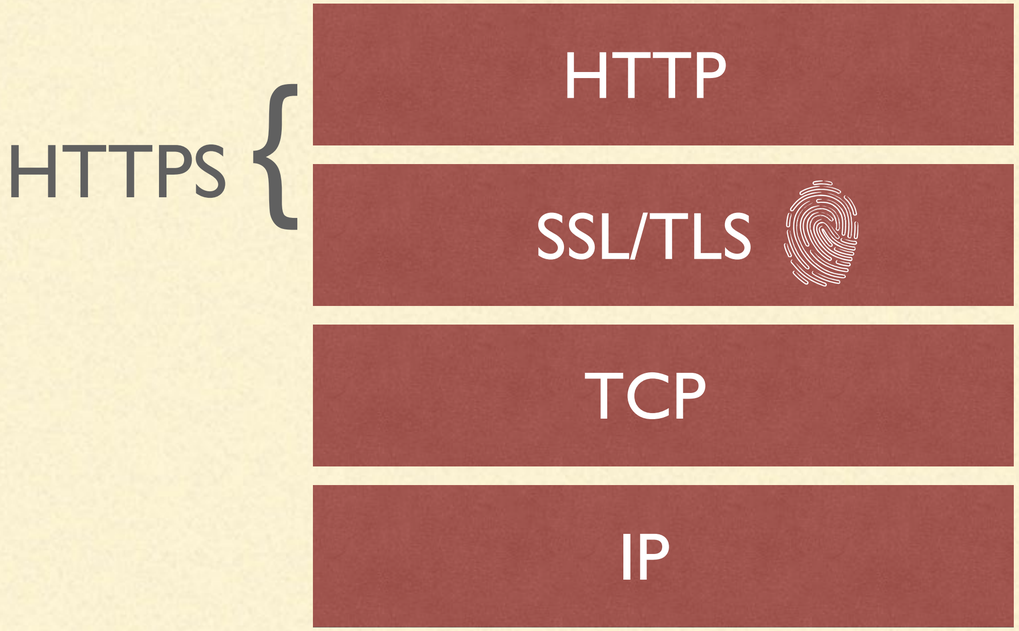
# SSL/TLS协议简介

Ding@18.11

# 什么是SSL/TLS

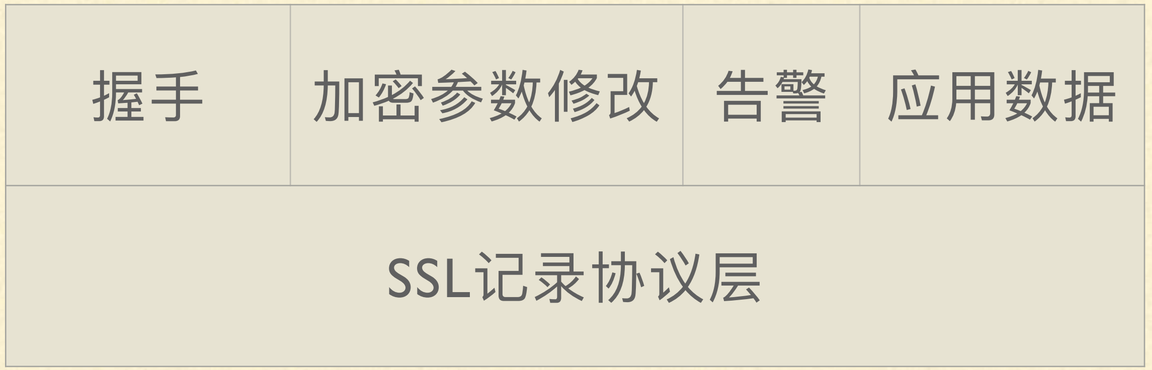
SSL(SECURE SOCKET LAYER)是NETSCAPE公司设计的主要用于WEB的安全传输协议。
IETF将SSL作了标准化(RFC2246)并将其称为TLS(TRANSPORT LAYER SECURITY)。

SSL/TLS的目标是：
1. 认证用户和服务器，确保数据发送到正确的客户机和服务器
2. 加密数据以防止数据中途被窃取
3. 维护数据的完整性，确保数据在传输过程中不被改变
4. 如何在安全和性能上权衡



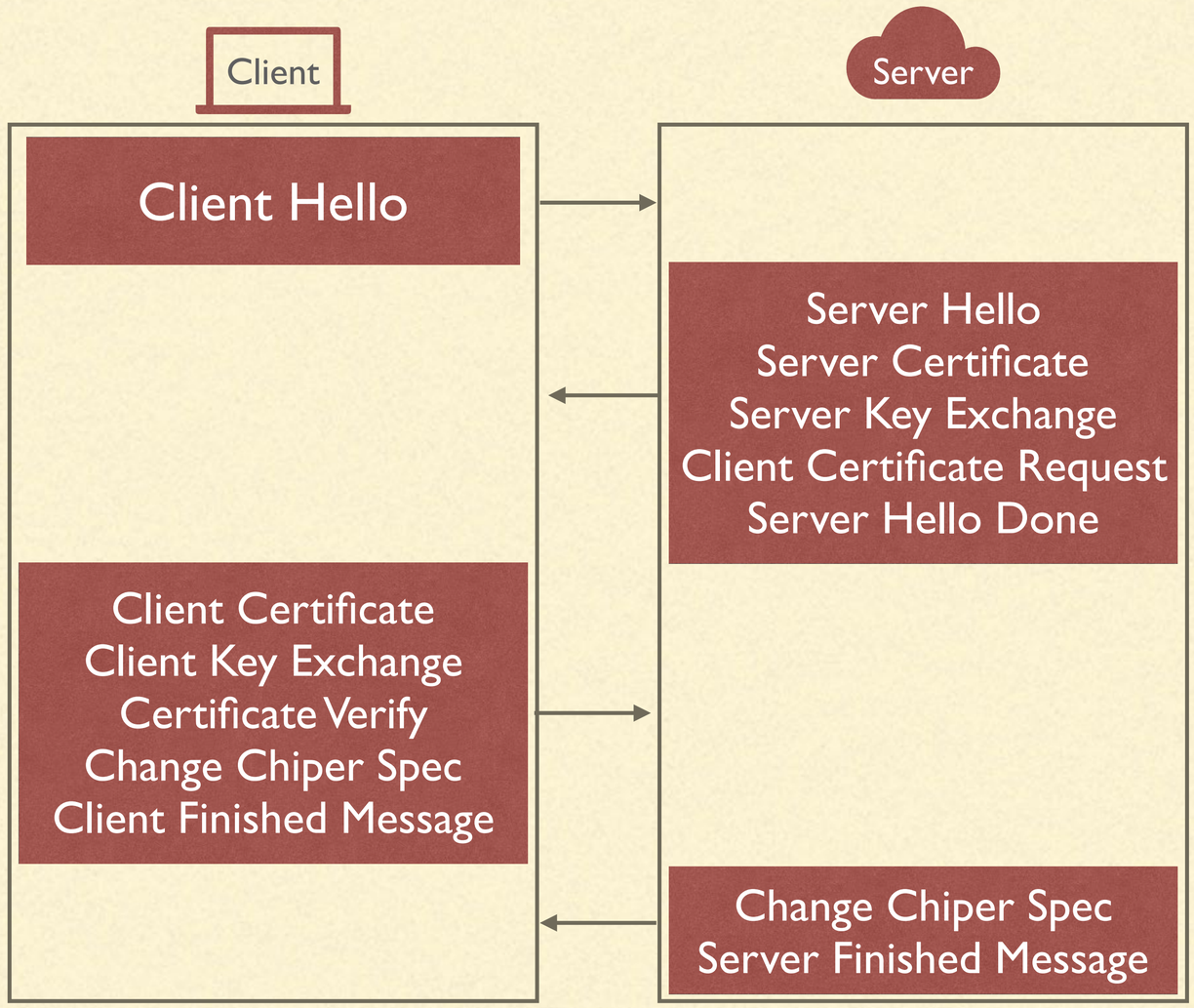
# 体系结构

- SSL Handshake Protocol Layer
- SSL Record Protocol Layer



# SSL Handshake Protocol

- Initial Client Message to Server
- Server Response to Client
- Client Response to Server
- Server Final Response to Client



# SSL Handshake Protocol

## Initial Client Message To Server

Client Hello

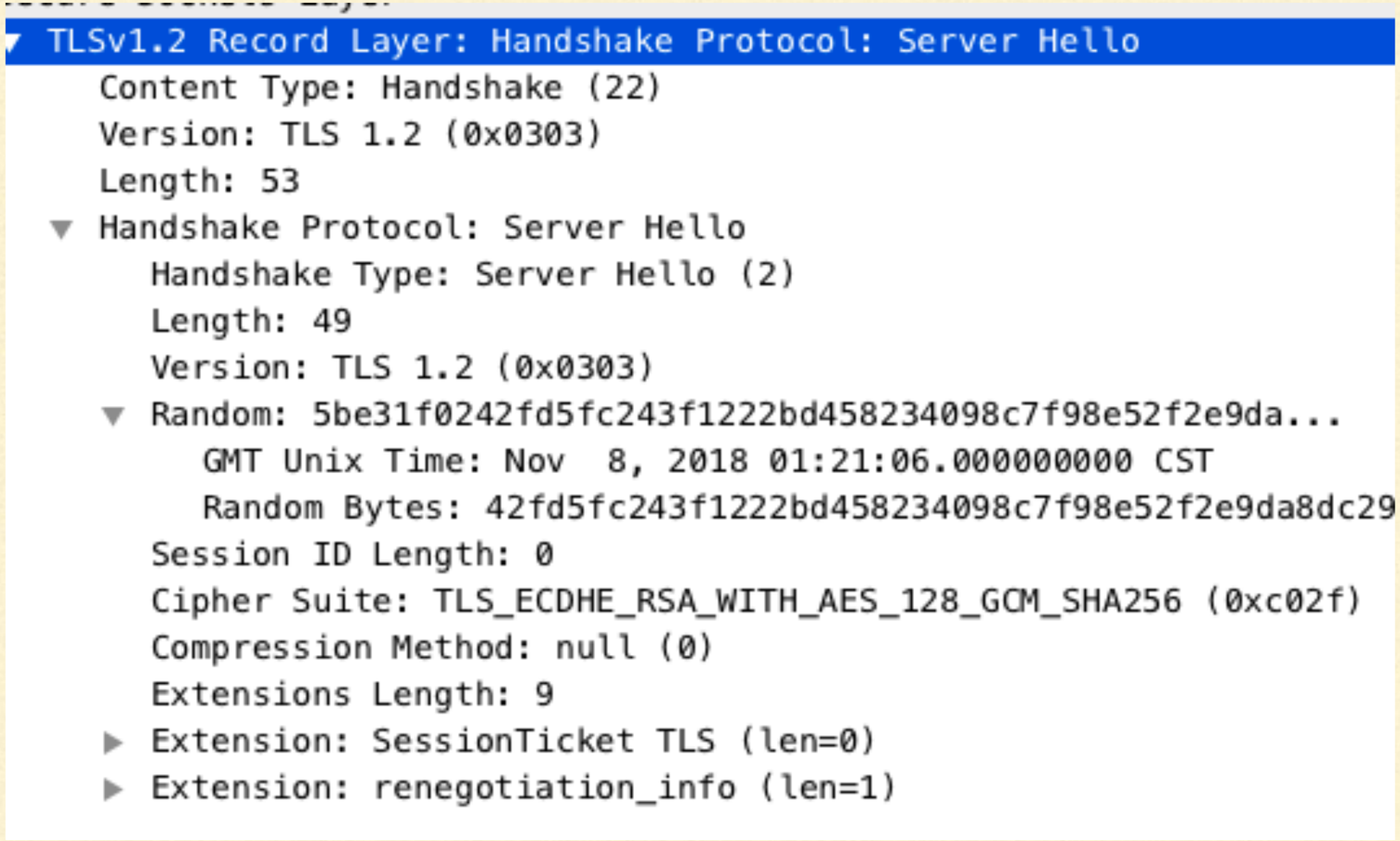
- Version Number
- Randomly Generated Data
- Session Identification(Session ID)
- Chiper Set
- Compression Algorithm

```
▼ TLSv1.2 Record Layer: Handshake Protocol: Client Hello
    Content Type: Handshake (22)
    Version: TLS 1.0 (0x0301)
    Length: 195
  ▼ Handshake Protocol: Client Hello
      Handshake Type: Client Hello (1)
      Length: 191
      Version: TLS 1.2 (0x0303)
    ▶ Random: 900e15a814bfaff8c43e43a6e12c18c9c051cffa736d6
      Session ID Length: 0
      Cipher Suites Length: 96
    ▶ Cipher Suites (48 suites)
      Compression Methods Length: 1
    ▶ Compression Methods (1 method)
      Extensions Length: 54
    ▶ Extension: ec_point_formats (len=2)
    ▶ Extension: supported_groups (len=8)
    ▶ Extension: SessionTicket TLS (len=0)
    ▶ Extension: signature_algorithms (len=28)
```

# SSL Handshake Protocol

## Server Response to Client

- Server Hello
  - Version Number
  - Randomly Generated Data
  - Session Identification
  - Cipher Suite
  - Compression Algorithm

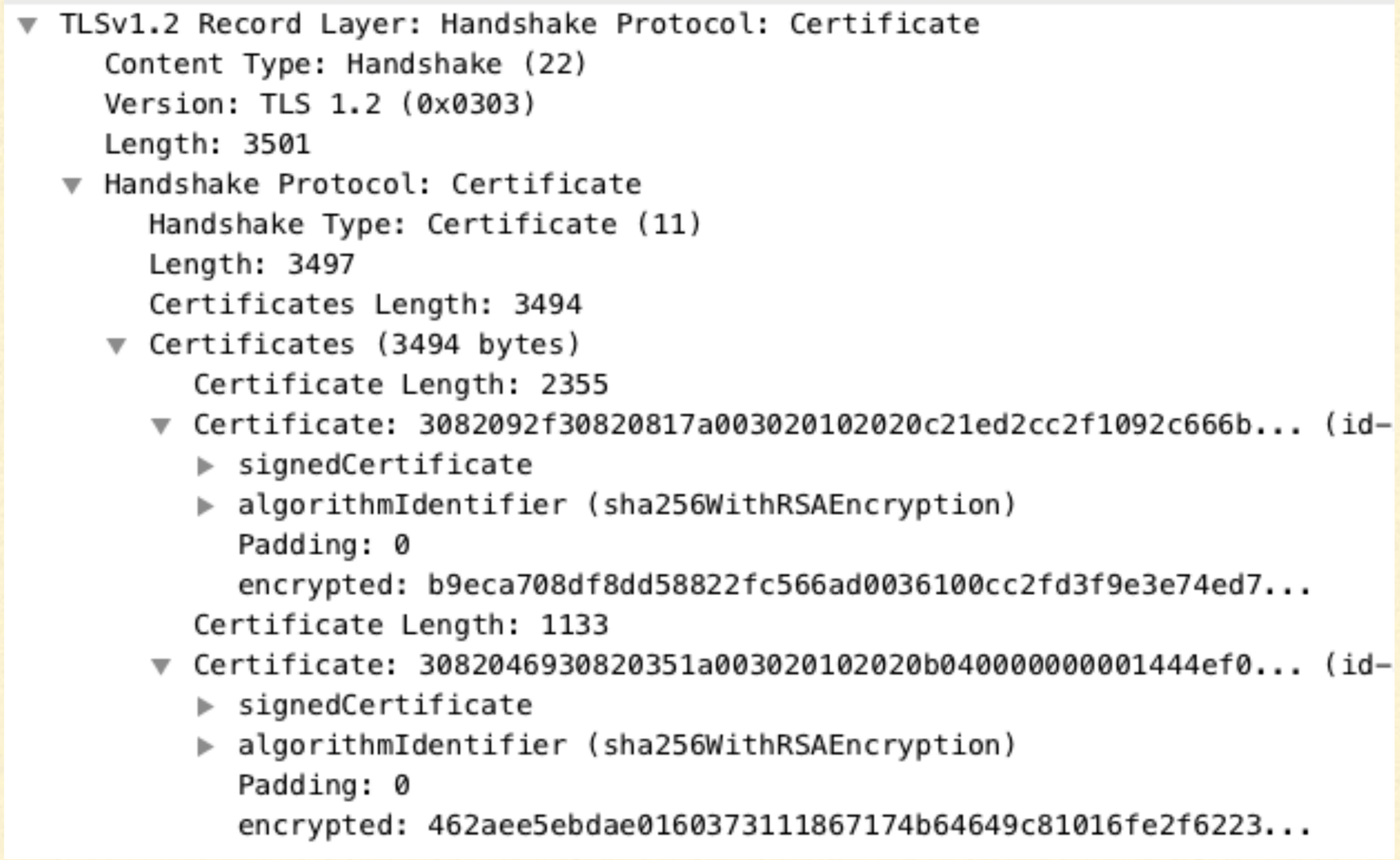


# SSL Handshake Protocol

## Server Response to Client

### Server Certificate

Server把自身的证书链发送给client。证书链中包含服务端的公钥。证书链的校验详阅PKI过程

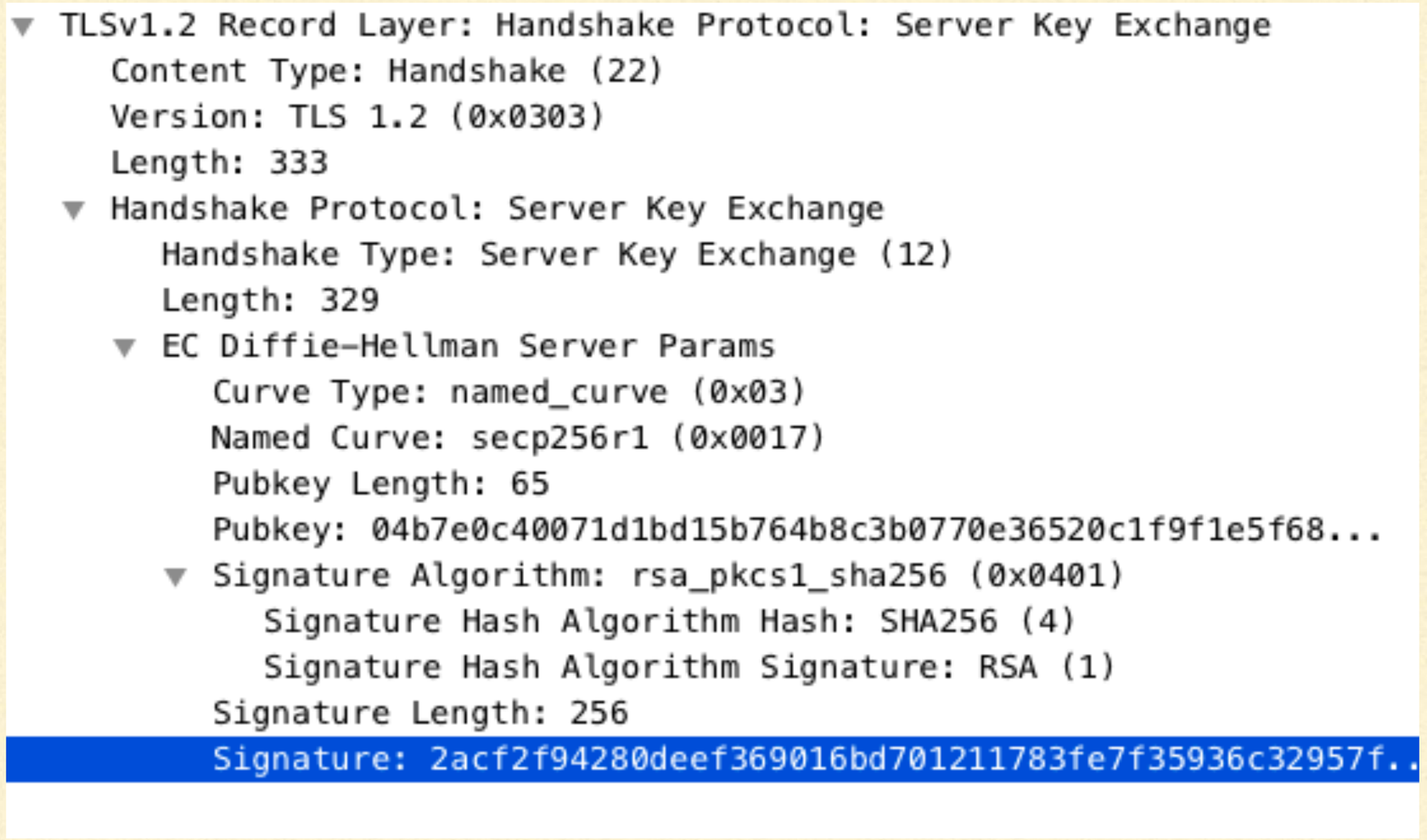


# SSL Handshake Protocol

## Server Response to Client

### Server Key Exchange

这一步是可选的，当上一步的证书中未能提供对应的支持客户端加密的算法时，才会产生这一步



# SSL Handshake Protocol

## Server Response to Client

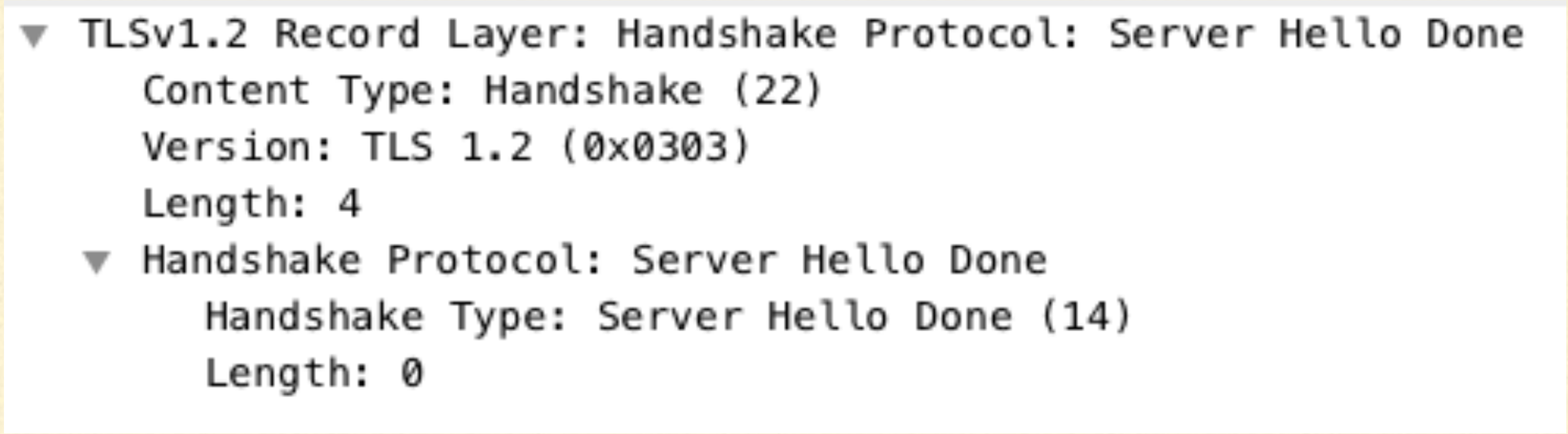
### Client Certificate Request

这一步是可选的，在一些Server需要验证client合法性的时候会用到。一般用在敏感信息的交互。例如：银行网站

# SSL Handshake Protocol

## Server Response to Client

### Server Hello Done



# SSL Handshake Protocol

## Client Response to Server

### Client Certificate

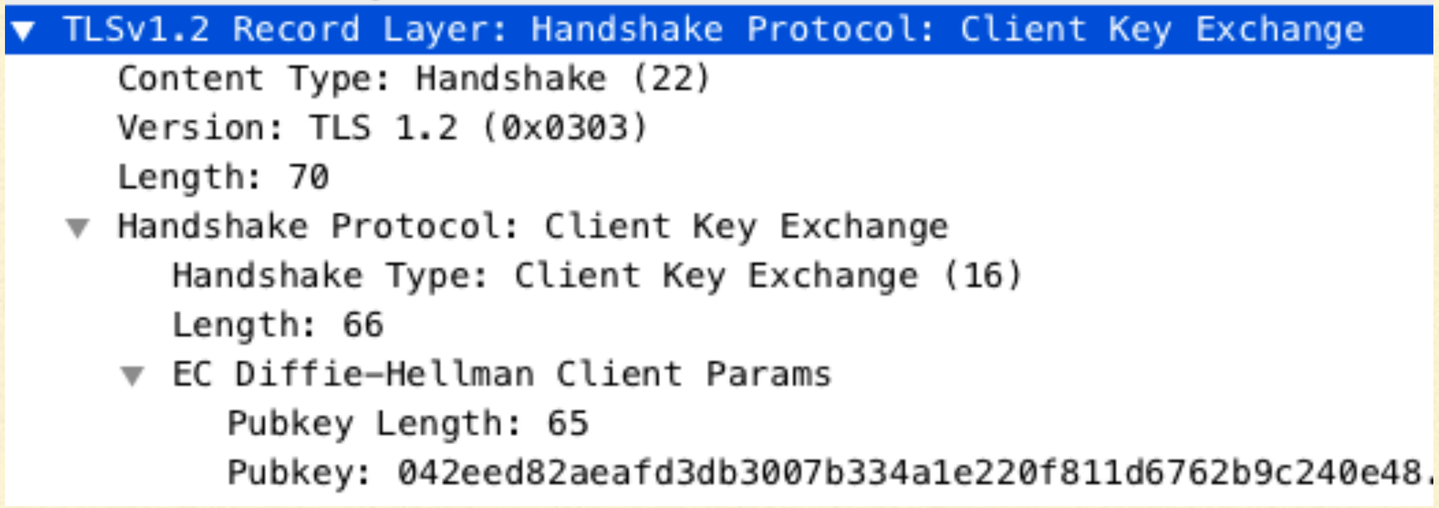
如果客户端收到了Client Certificate Request，客户端会接下来把自己的证书发送给Server。该证书包含客户端自己的公钥。

# SSL Handshake Protocol

## Client Response to Server

### Client Key Exchange

Client用Server的公钥加密pre-master之后，发送给Server



# SSL Handshake Protocol

## Client Response to Server

### Client Key Exchange

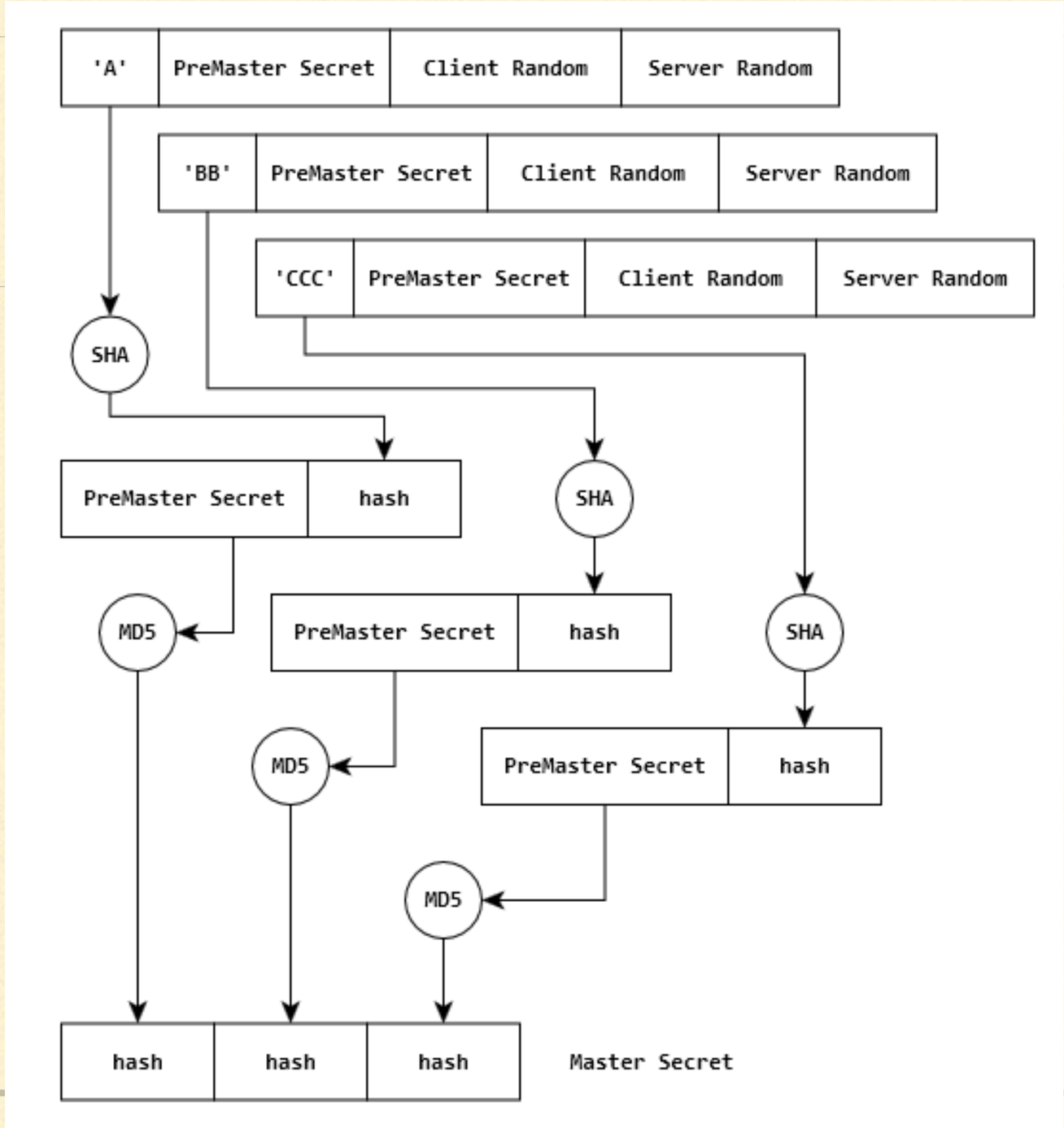
比较通用的一种情况就是：Client Random 和 Server Random 都获取到之后，在这一步生成一个新的随机数 pre-master-secret，然后使用数字证书中的公钥来加密 pre-master-secret 并发送给Server。Server会用私钥来解开公钥加密过的pre-master-secret. 然后结合之前双方都知道的两个Random，用PRF算法来生成master_secret。

选用了不同加密方法的Key交换的过程略有不同。

# SSL Handshake Protocol

## Client Response to Server

### Client Key Exchange



# SSL Handshake Protocol

## Client Response to Server

### Certificate Verify

仅当客户端先前发送了客户端证书消息时才会发送此消息。用来验证客户端证书的合法性。

# SSL Handshake Protocol

## Client Response to Server

```
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
```

### Change Cipher Spec

告知Server Client Finished消息之后的所有的消息都会采用之前协商的keys以及加密算法来进行通信

### Client Finished

# SSL Handshake Protocol

## Server Final Response to Client

Change Cipher Spec

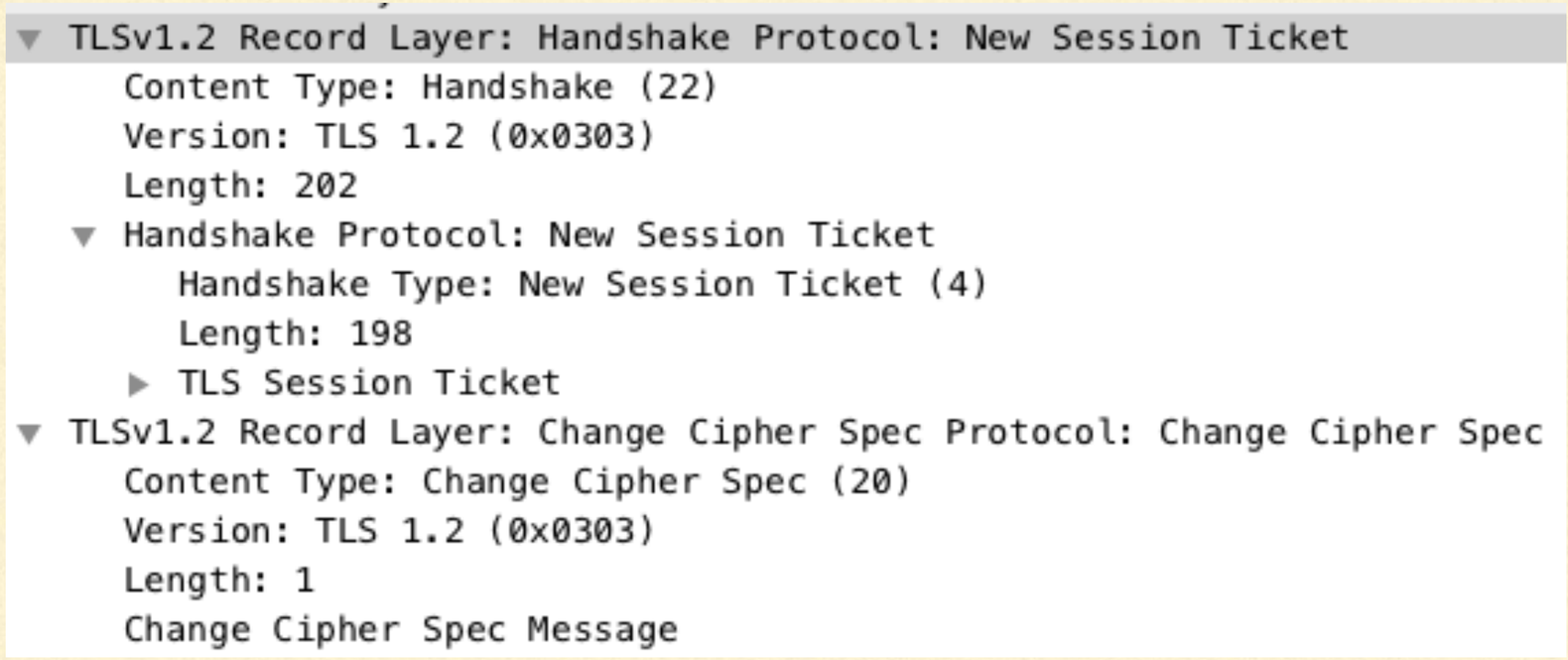
告知Client之后的所有

的消息都会采用之前协

商的keys以及加密算法

来进行通信

Server Finished



# SSL Record Protocol

## Session Keys

Session Keys由master secret经过RFC规定的PRF方法（可简单理解多hash多轮&拼接直到达到指定的长度）来生成，Session Keys包含至少4个密钥（也可能是6个）。

Client Write Mac(Hmac) Secret
Server Write Mac(Hmac) Secret
Server Write Key
Client Write Key
Server Write IV *
Client Write IV *

# TLS VS SSL

- 1）版本号：TLS记录格式与SSL记录格式相同，但版本号的值不同，TLS的版本1.0使用的版本号为SSLv3.1。
- 2）报文鉴别码：SSLv3.0和TLS的MAC算法及MAC计算的范围不同。TLS使用了RFC-2104定义的HMAC算法。SSLv3.0使用了相似的算法，两者差别在于SSLv3.0中，填充字节与密钥之间采用的是连接运算，而HMAC算法采用的是异或运算。但是两者的安全程度是相同的。
- 3）伪随机函数：TLS使用了称为PRF的伪随机函数来将密钥扩展成数据块，是更安全的方式。
- 4）报警代码：TLS支持几乎所有的SSLv3.0报警代码，而且TLS还补充定义了很多报警代码，如解密失败（decryption_failed）、记录溢出（record_overflow）、未知CA（unknown_ca）、拒绝访问（access_denied）等。
- 5）密文族和客户证书：SSLv3.0和TLS存在少量差别，即TLS不支持Fortezza密钥交换、加密算法和客户证书。
- 6）certificate_verify和finished消息：SSLv3.0和TLS在用certificate_verify和finished消息计算MD5和SHA-1散列码时，计算的输入有少许差别，但安全性相当。
- 7）加密计算：TLS与SSLv3.0在计算主密值（master secret）时采用的方式不同。
- 8）填充：用户数据加密之前需要增加的填充字节。在SSL中，填充后的数据长度要达到密文块长度的最小整数倍。而在TLS中，填充后的数据长度可以是密文块长度的任意整数倍（但填充的最大长度为255字节），这种方式可以防止基于对报文长度进行分析的攻击。

永远相信美好的事情即将发生

*Always believe that something wonderful is about to happen*