网络安全协议及分析
传输层安全SSL和TLS

密码与网络安全系　刘虹

2025年春季学期

课程体系

第一章 概述

第二章 链路层扩展L2TP

第三章 IP层安全IPSec

第四章 传输层安全SSL和TLS

第五章 会话安全SSH

第六章 代理安全Socks

第七章 网管安全SNMPv3

第八章 认证协议Kerberos

第九章 应用安全

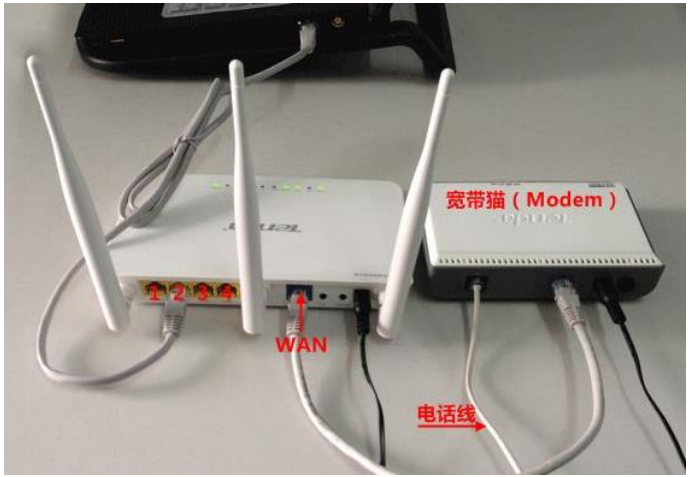
本章学习目标

- SSL协议概述
- SSLv3协议的流程
- 密钥导出机制

提纲

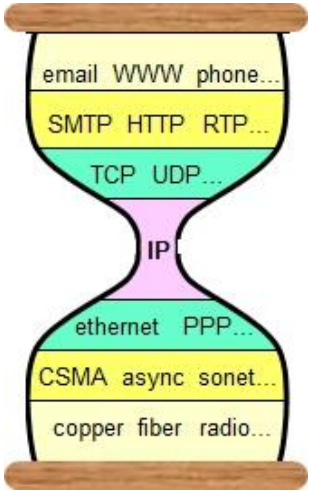
链路层安全

- 第二层隧道协议L2TP
  - 对网络接口层PPP协议（Point to Point Protocol）进行了扩展，是的用户可以通过互联网建立一条点到点链路。
  - 用于两个对等实体之间的直连链路，这种链路使用**拨号或专线**连接方式，提供全双工的数据传输服务，数据按序传输。



IP层安全

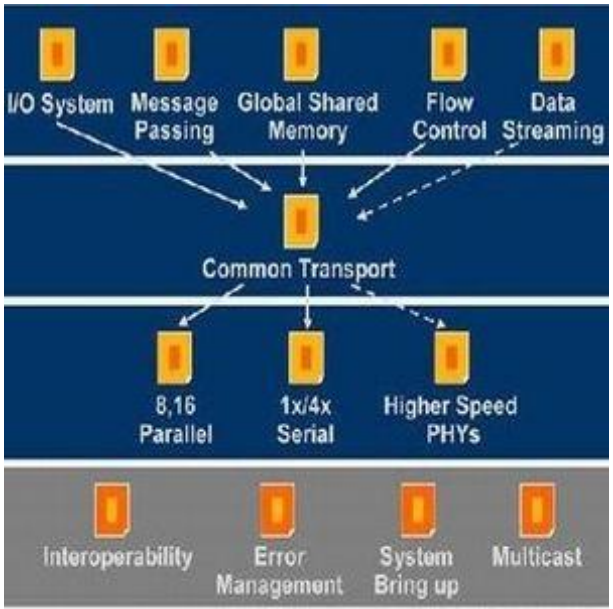
- 交换协议：
  - 互联网安全关联与密钥管理协议（Internet Security Association and Key Management Protocol，lSAKMP)
  - 互联网密钥交换（Internet Key Exchange，IKE)
- 数据封装协议：
  - 数据封装处理协议的认证首部（Authentication Header，AH)
  - 封装安全载荷（Encapsulating Security Payload，ESP)
- IPSec协议优势：
  - IPsec在通用性
  - 部署灵活性
  - 安全策略统一性



传输层安全

- 传输层在TCP/IP协议族中具有重要地位，加强和弥补了IP层的服务
  - 端到端服务
  - 应用层直接构建于传输层之上

- 安全套接层（Secure Socket Layer，SSL）和传输层安全（Transport Layer Security，TLS）



SSL协议

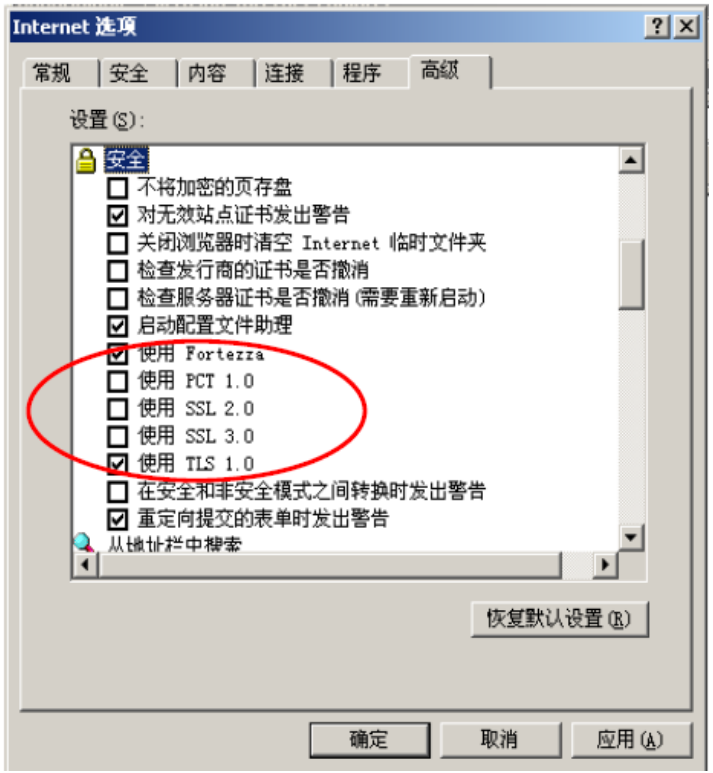
- 设计目的
  - 增强HTTP的安全性
  - 让普通的HTTP协议和增强安全性的协议之间具有良好的互操作性
- 安全需求
  - 机密性
  - 完整性
  - 服务器身份认证
  - 可选的用户身份认证

SSL协议

- 1994年Netscape开发SSL协议，专门用于保护Web通信
  - 1.0，不成熟
  - 2.0，基本上解决Web安全问题
- Microsoft发布 Private Communication Technology（PCT），并在IE中支持
  - 3.0，1996年发布，增加算法，修改缺陷
  - TLS 1.0（被称为SSL 3.1）
  - 1999年，发布RFC 2246

SSL协议

- SSL协议
  - 保护运行于TCP之上的应用协议。
  - 与应用层协议无关的。
  - 在应用层协议之前就已经完成加密算法、通信密钥的协商以及服务器的认证工作。



提纲

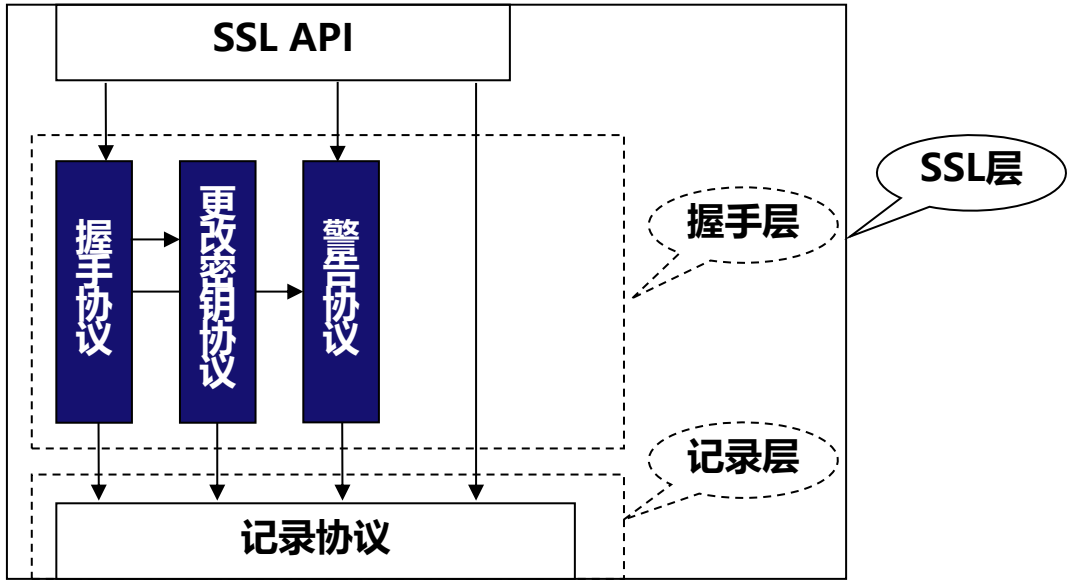
一、SSL协议概述

二、SSLv3协议的流程

三、密钥导出机制

二 SSL协议

- 从协议栈层次关系看，SSL位于应用层和传输层之间
- SSLv3是一个协议套件
  1. 握手协议
  2. 记录协议
  3. 更改密码规范协议
  4. 警告协议



二 SSL协议

握手协议

- 必选的服务器认证，即客户端认证服务器的身份。
- 可选的客户端认证，即服务器可以选择认证客户端的身份，也可以放弃该功能。
- 算法协商，即客户端和服务器协定采用的压缩算法、加密算法和消息验证码算法。
- 密钥生成，即生成客户端和服务器共享的会话密钥。

二 SSL协议

记录协议

- SSLv3的数据承载协议，规定了SSLv3的报文格式以及对报文的处理过程。
- 握手报文及应用数据都要封装成“记录”的形式投递。

更改密码规范协议

- 只包括一条消息。
- 在安全协商完成后，客户端和服务器会交换这条消息，以通告对方启用协商好的安全参数，随后的消息都将用这些参数保护。

二 SSL协议

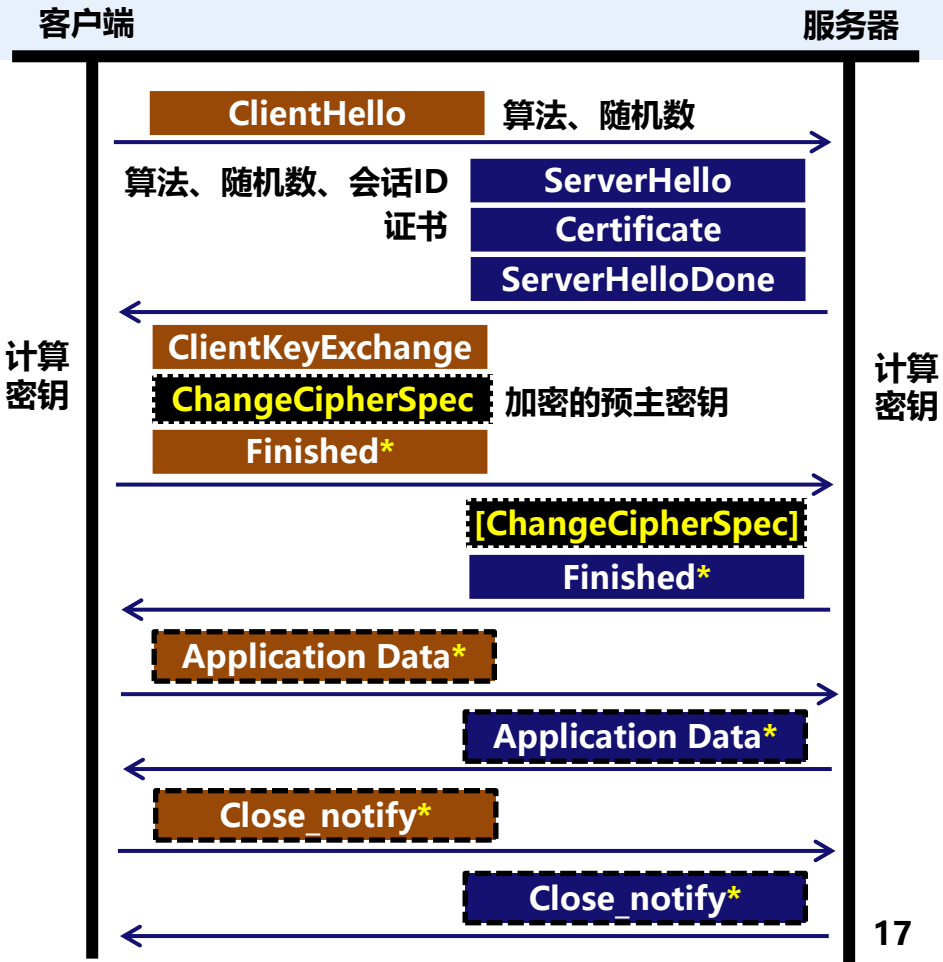
- **警告协议**
  - **提供报错机制**，即通信过程中若某一方发现了问题，利用该协议通告对等端；
  - **提供安全断连机制**，即以一种可认证的方式关闭连接。TCP是面向连接的，使用不加认证的断连方式会面临“截断"攻击的威胁，安全断连用于解决该问题。

二 SSL协议流程

- 典型的SSLv3通信过程：
  - 在传输应用数据之前，必须首先进行协商，以便客户端验证服务器的身份，并与服务器就算法和密钥达成共识；
  - 在数据传输阶段，双方利用协商好的算法和密钥处理应用数据；
  - 数据传输完成后，通过可认证的方式断开连接。

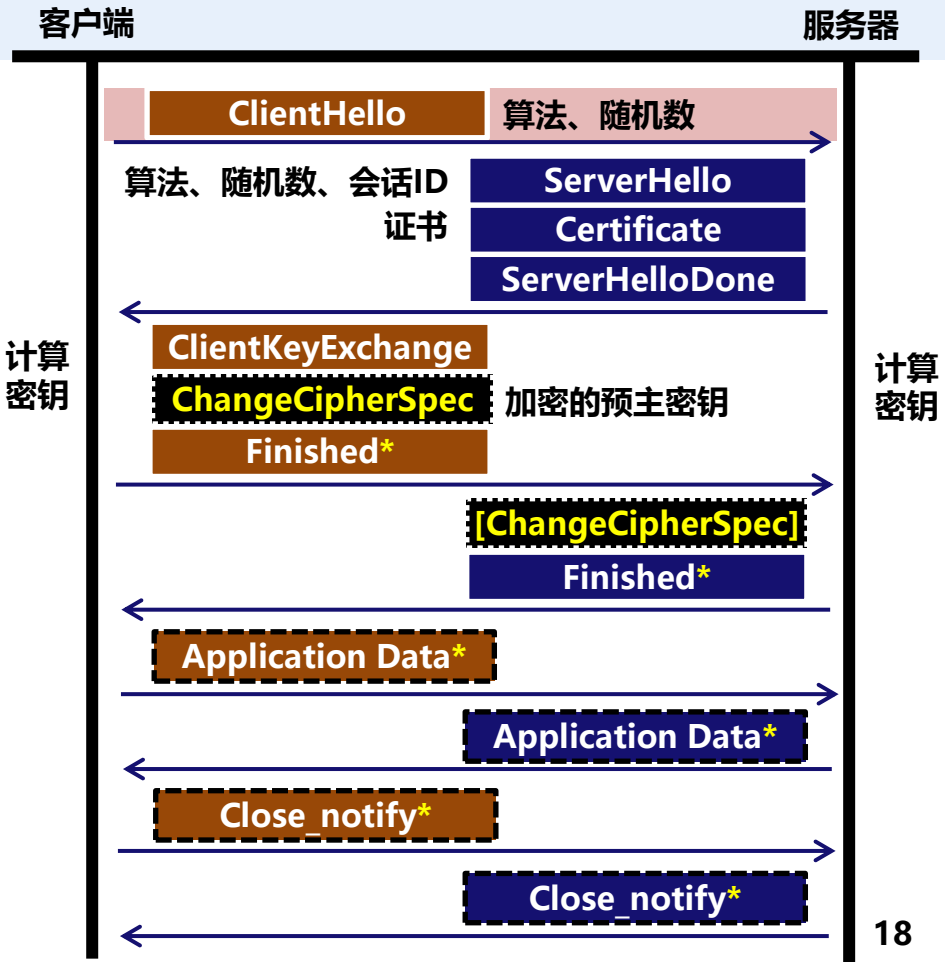
二 SSL协议流程

1. 客户端向服务器发送ClientHello消息
2. 服务器返回ServerHello消息
3. 服务器返回Certificate消息
4. 服务器发送ServerHelloDone消息
5. 客户端向服务器发送ClientKeyExchange消息
6. 客户端和服务器以预主密钥和随机数作为输入
7. 客户端向服务器发送ChangeCipherSpec消息
8. 服务器向客户端发送ChangeCipherSpec消息
9. 服务器向客户端发送Finished消息
10. 客户端和服务器之间交互应用数据
11. 客户端向服务器发送Close_notify消息
12. 服务器向客户端发送Close_notify消息



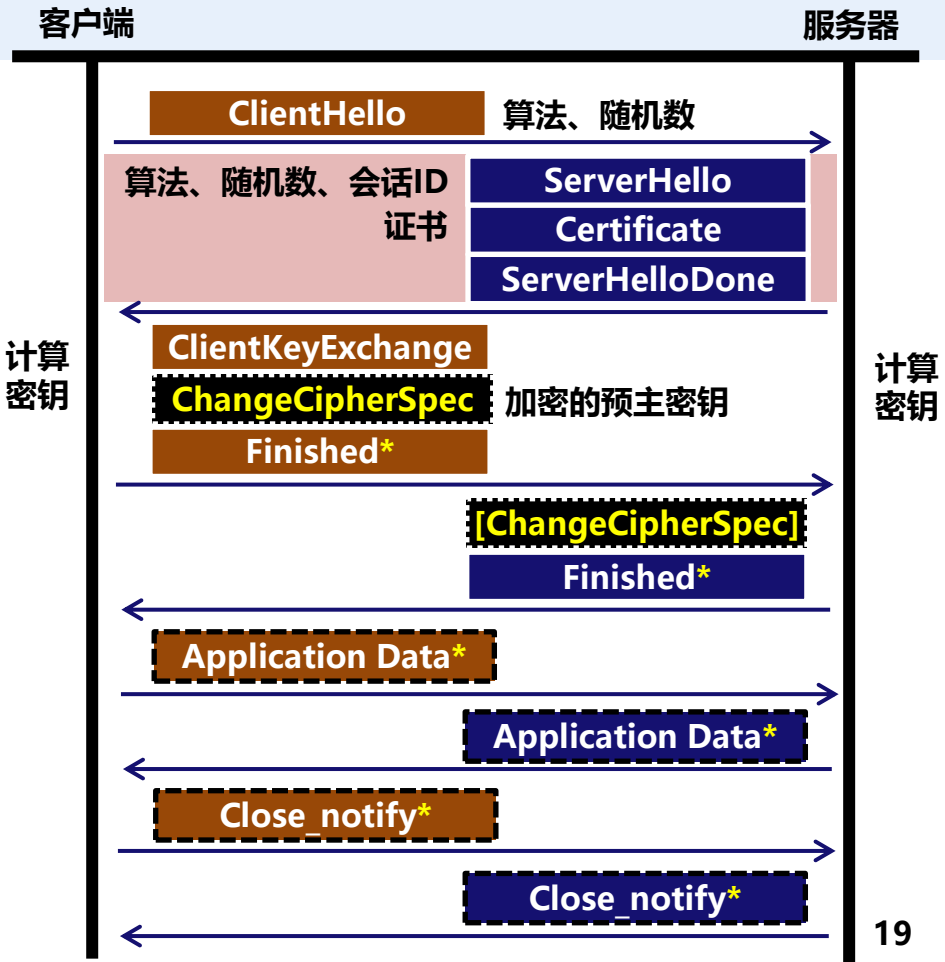
二 SSL协议流程

1. 客户端向服务器发送ClientHello消息
2. 服务器返回ServerHello消息
3. 服务器返回Certificate消息
4. 服务器发送ServerHelloDone消息
5. 客户端向服务器发送ClientKeyExchange消息
6. 客户端和服务器以预主密钥和随机数作为输入
7. 客户端向服务器发送ChangeCipherSpec消息
8. 服务器向客户端发送ChangeCipherSpec消息
9. 服务器向客户端发送Finished消息
10. 客户端和服务器之间交互应用数据
11. 客户端向服务器发送Close_notify消息
12. 服务器向客户端发送Close_notify消息



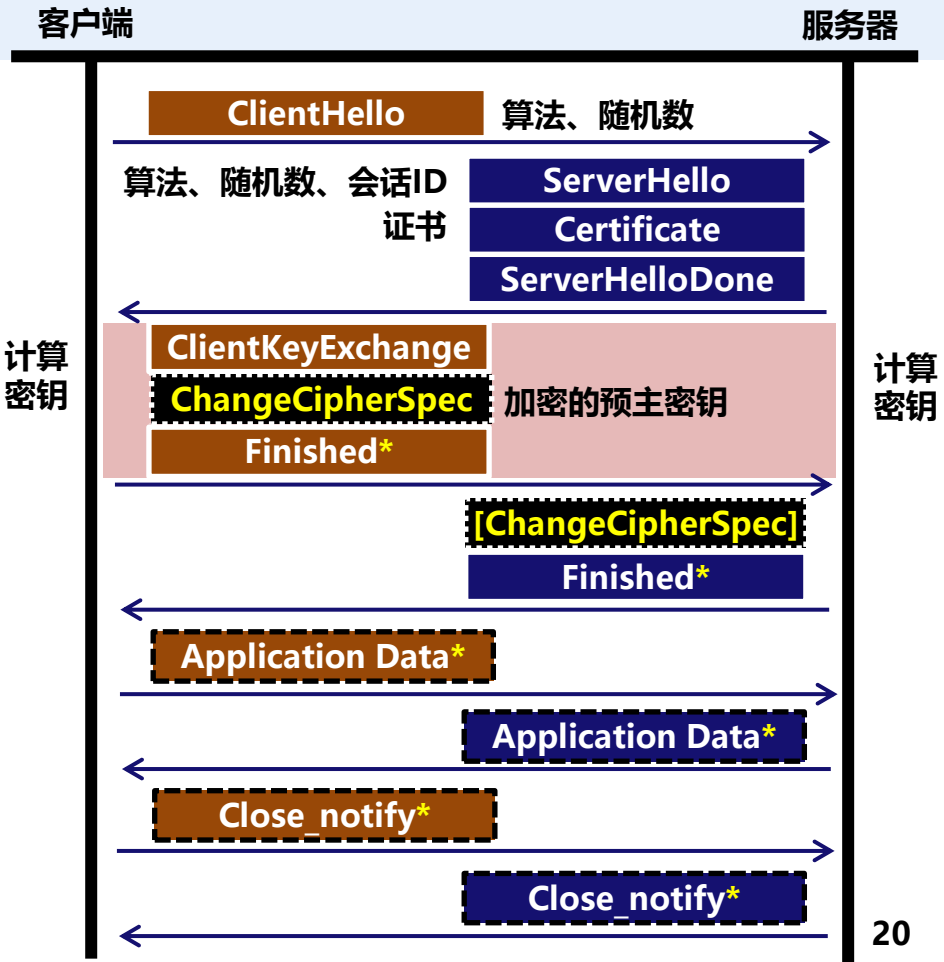
二 SSL协议流程

1. 客户端向服务器发送ClientHello消息
2. **服务器返回ServerHello消息**
3. **服务器返回Certificate消息**
4. **服务器发送ServerHelloDone消息**
5. 客户端向服务器发送ClientKeyExchange消息
6. 客户端和服务器以预主密钥和随机数作为输入
7. 客户端向服务器发送ChangeCipherSpec消息
8. 服务器向客户端发送ChangeCipherSpec消息
9. 服务器向客户端发送Finished消息
10. 客户端和服务器之间交互应用数据
11. 客户端向服务器发送Close_notify消息
12. 服务器向客户端发送Close_notify消息



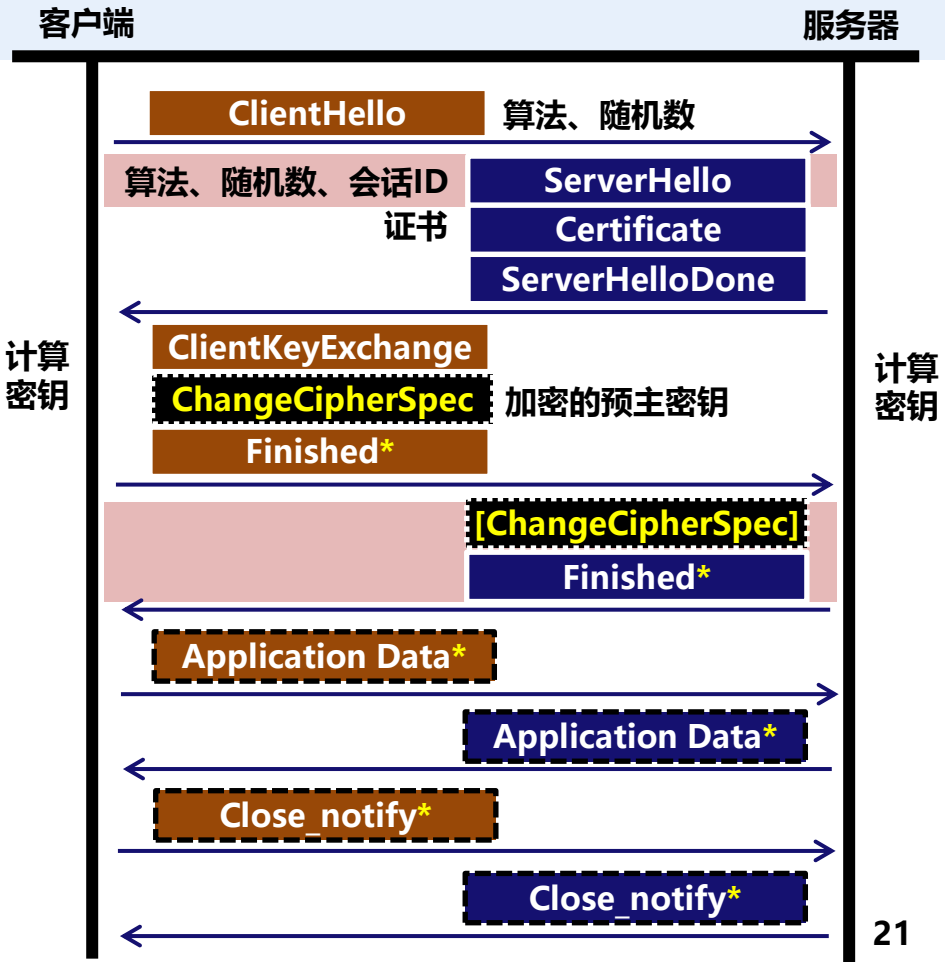
二 SSL协议流程

1. 客户端向服务器发送ClientHello消息
2. 服务器返回ServerHello消息
3. 服务器返回Certificate消息
4. 服务器发送ServerHelloDone消息
5. **客户端向服务器发送ClientKeyExchange消息**
6. **客户端和服务器以预主密钥和随机数作为输入**
7. **客户端向服务器发送ChangeCipherSpec消息**
8. 服务器向客户端发送ChangeCipherSpec消息
9. 服务器向客户端发送Finished消息
10. 客户端和服务器之间交互应用数据
11. 客户端向服务器发送Close_notify消息
12. 服务器向客户端发送Close_notify消息



二 SSL协议流程

1. 客户端向服务器发送ClientHello消息
2. 服务器返回ServerHello消息
3. 服务器返回Certificate消息
4. 服务器发送ServerHelloDone消息
5. 客户端向服务器发送ClientKeyExchange消息
6. 客户端和服务器以预主密钥和随机数作为输入
7. 客户端向服务器发送ChangeCipherSpec消息
8. **服务器向客户端发送ChangeCipherSpec消息**
9. **服务器向客户端发送Finished消息**
10. 客户端和服务器之间交互应用数据
11. 客户端向服务器发送Close_notify消息
12. 服务器向客户端发送Close_notify消息



二 SSL协议流程：拟人版

- 假设客户端与服务器通信，加密后的消息放在**方括号[]**里，以区别于明文消息。双方的处理动作的说明用**圆括号()**括起。
- 客户端：
  - 我想和你安全的通话，我这里的对称加密算法有DES、RC5分组密码算法，密钥交换算法有RSA和D-H，摘要算法有MD5和SHA。
- 服务器：
  - 我们用“**DES-RSA-SHA**”这对组合好了。
  - 这是我的**证书**，里面有我的名字和公钥，你拿去验证一下我的身份（把证书发给客户端）。

二 SSL协议流程：拟人版

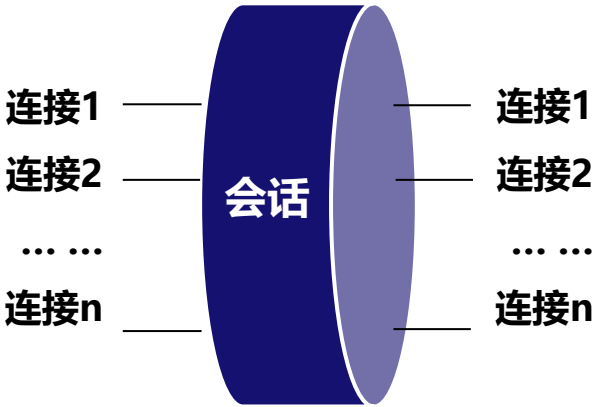
- 客户端：
  - （查看证书上服务器的名字，并通过手头已有的CA的证书验证服务器的证书的真实性；如果有误，发出警告并断开连接，这一步保证了公钥的真实性）
  - （产生一份秘密消息——称为per_master_secret ，以后用于对初始化向量和hmac进行加密，然后用服务器的公钥加密成为ClientKeyExchange）
  - 我生成了一份秘密消息，并用你的公钥加密传给你了
  - （把ClientKeyExchange发给服务器）
  - 注意：下面我要用加密方法给你发消息了！
  - （将秘密消息进行处理，生成加密密钥，加密初始化向量和HMAC的密钥）
  - [我说完了]

二 SSL协议流程：拟人版

- 服务器：
  - 用自己的私钥对ClientKeyExchange解密，然后将秘密消息进行处理，生成加密密钥，加密初始化向量和HMAC的密钥，这时双方已经安全的协商出一套加密办法了）
  - 注意：我也要开始用加密办法给你发消息了！
  - [我说完了]
- 客户端：
  - [我的秘密是...]
- 服务器：
  - [其它人不会听到的...]

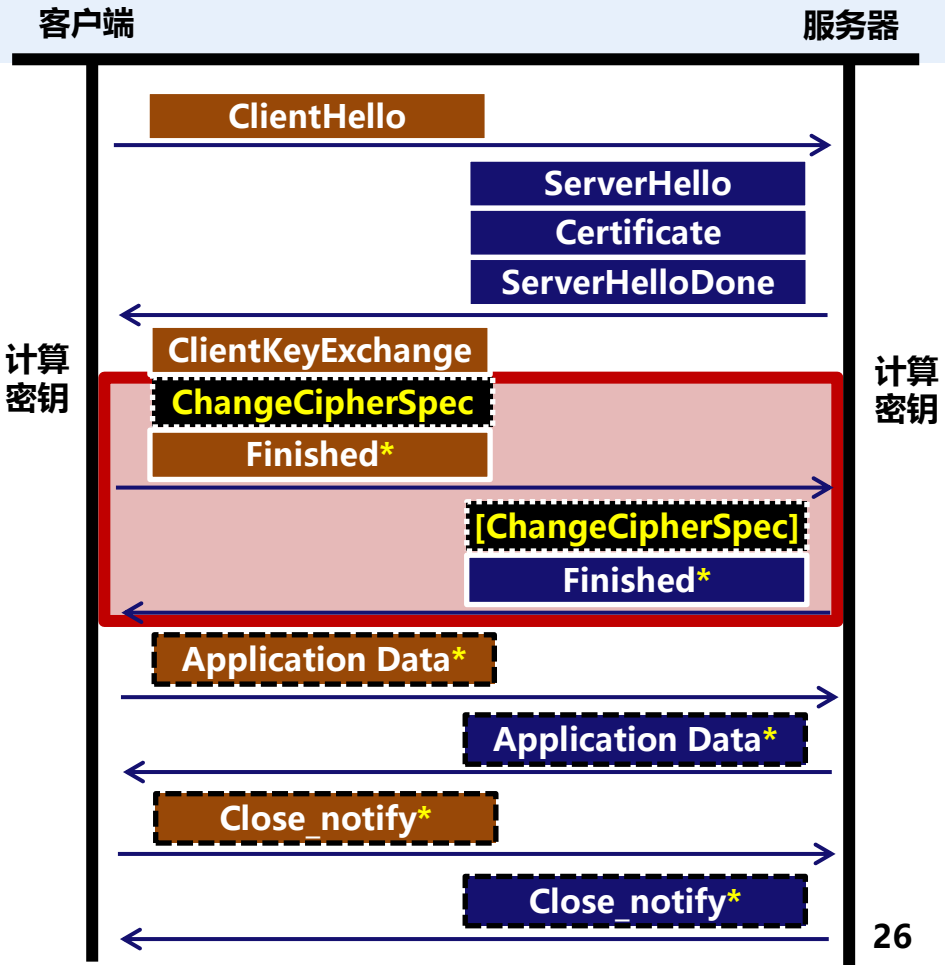
二 SSL协议流程

- 更改密码规范协议
  - SSL连接（Connection）
    - 一个连接是一个提供合适类型服务的传输。
    - SSL的连接是点对点的关系。
    - 连接是暂时的，每一个连接和一个会话关联。
  - SSL会话（Session）
    - 一个会话是在客户与服务器之间的一个关联。会话定义了一组可供多个连接共享的密码安全参数。
    - 会话用以避免为每一个连接提供新的安全参数所需昂贵的协商代价。



二 SSL协议流程

- 握手协议
  - 协调客户端和服务器的状态，使得通信双方的交互实现同步。
  - 当前操作状态和挂起状态分别表示正在使用和正在协商的密码参数。



二 SSL协议流程

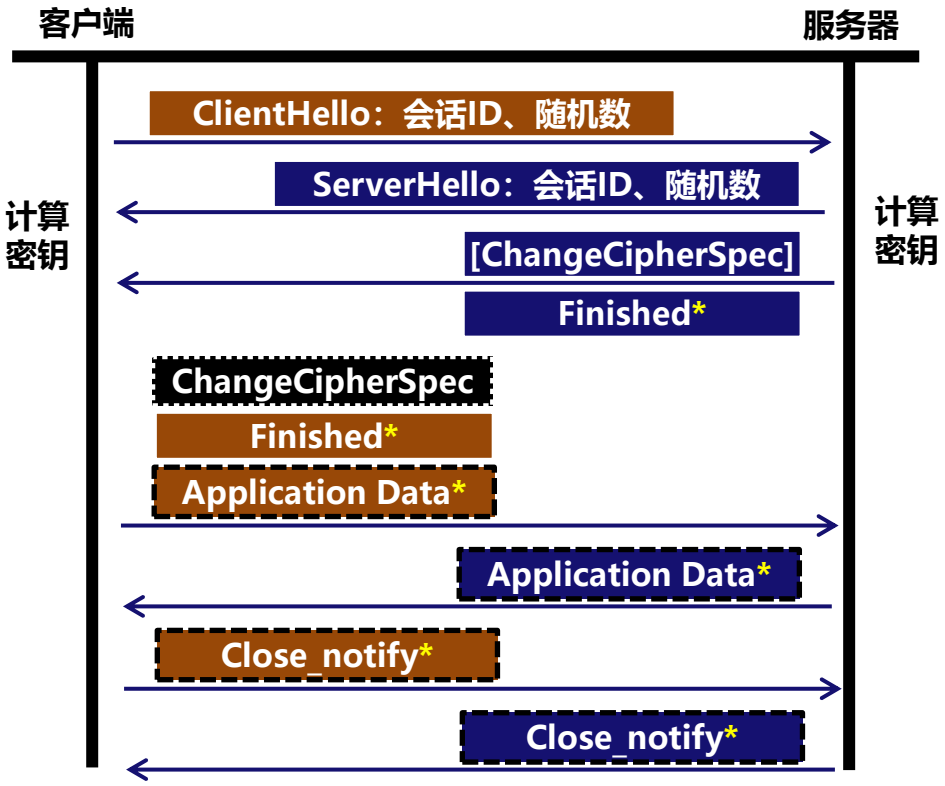
- 警告协议
  - 无论是在协商还是在应用数据传输阶段，如果通信一方发现了差错，必须向对方报告；应用数据传输完成后，还必须通知对方断开连接。
  - 警告级（Warning）、致命级（Fatal）
  - SSLv3的断连消息为Close_notify，引入消息验证码机制。

二 SSL协议流程

- 会话和连接
  - **会话状态**：一组参数，包括会话ID、通信对等端证书、压缩算法、加密算法、散列算法、预主密钥以及可恢复标记。
  - **连接状态**：一组参数，包括客户端和服务器随机数、服务器MAC密钥、客户端MAC密钥、服务器加密密钥、客户端加密密钥、初始化向量IV以及序号。

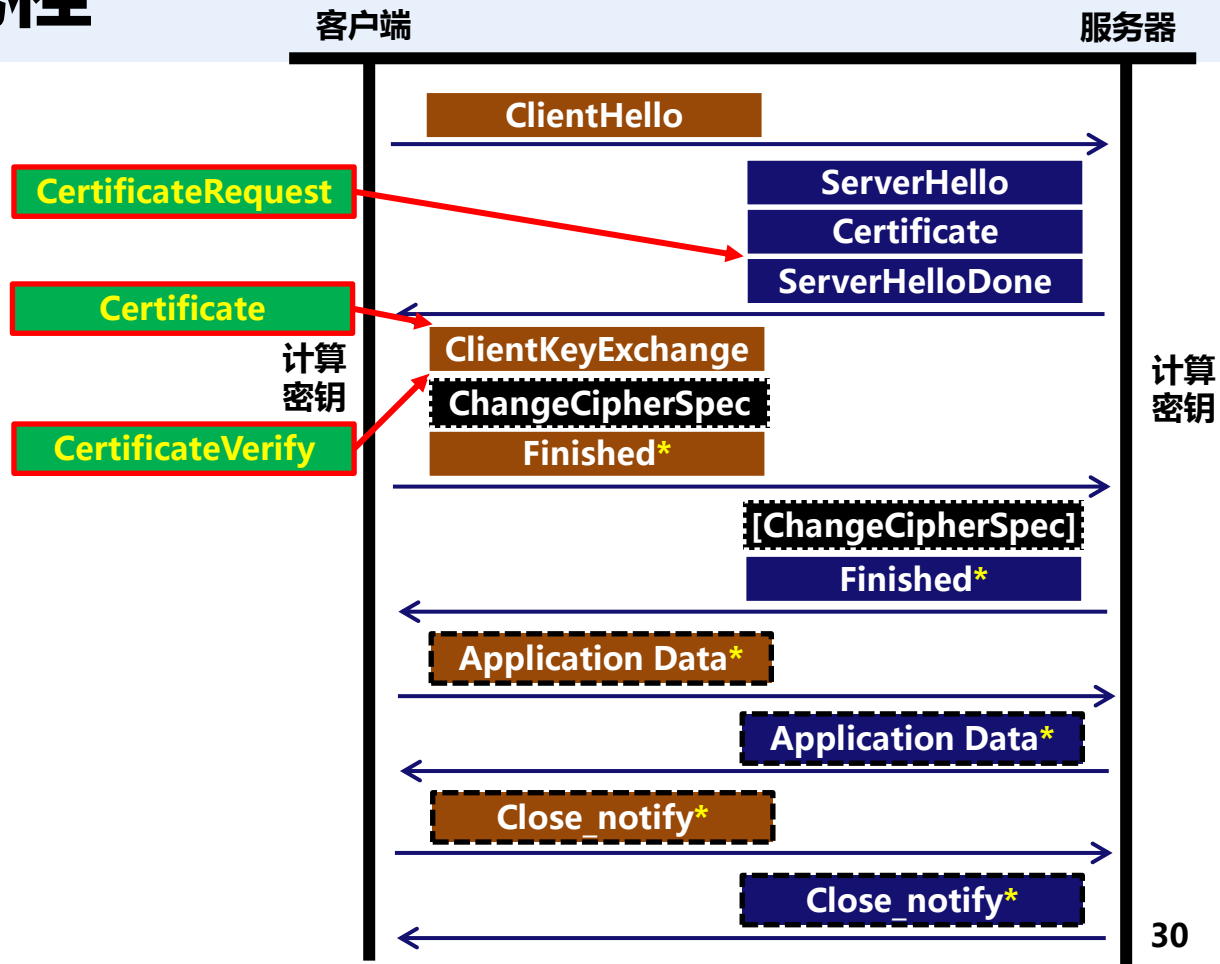
二 SSL协议流程

- 会话恢复
  - 使用已有的会话保护某个连接，不必再重新协商新参数。
  - 使用会话恢复可提高通信效率。



二 SSL协议流程

- 客户端认证
  - 服务器认证仅发送证书
  - 客户端认证多发送CertificateVerify消息。
  - 真正的服务器认证过程隐含在密钥生成过程中。



提纲

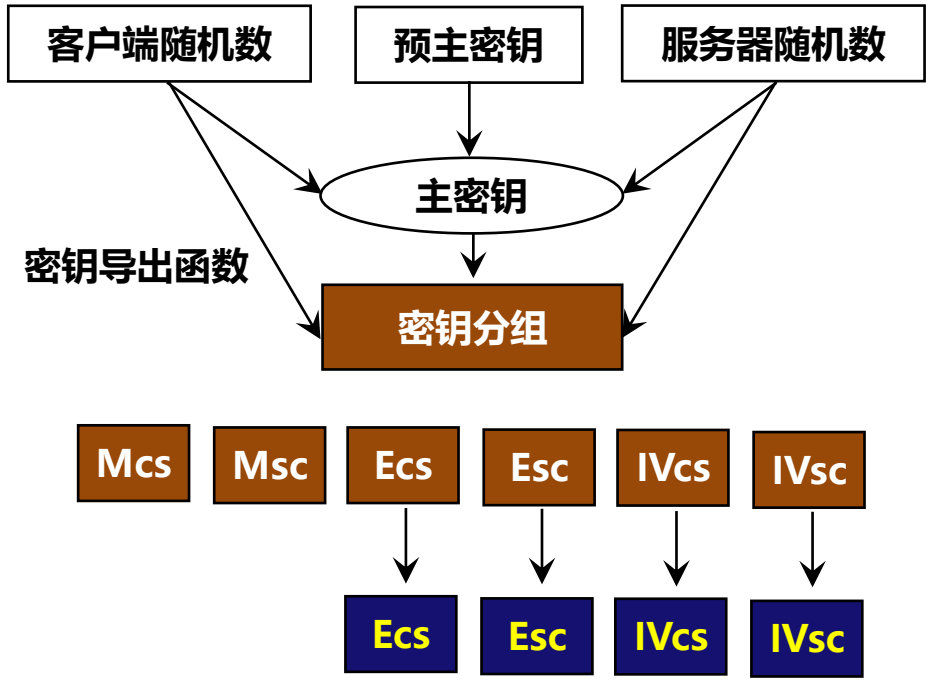
一、SSL协议概述

二、SSLv3协议的流程

三、密钥导出机制

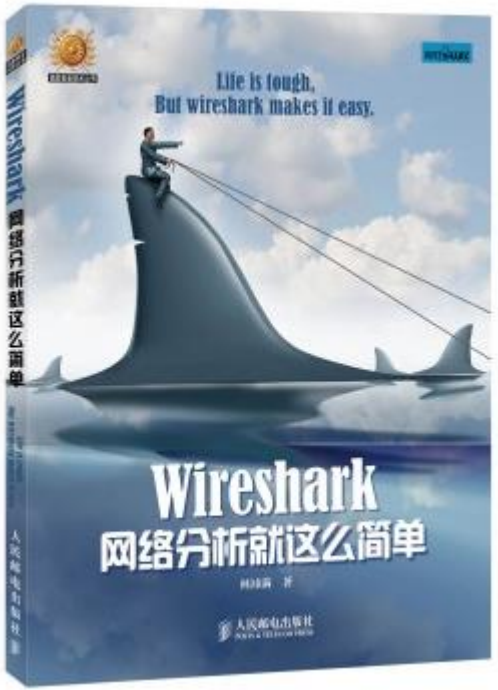
三 密钥导出

- 4个用于保护数据的会话密钥
  - Esc：服务器写加密密钥
  - Msc：服务器写MAC密钥
  - Ecs：客户端写加密密钥
  - Mcs：客户端写MAC密钥



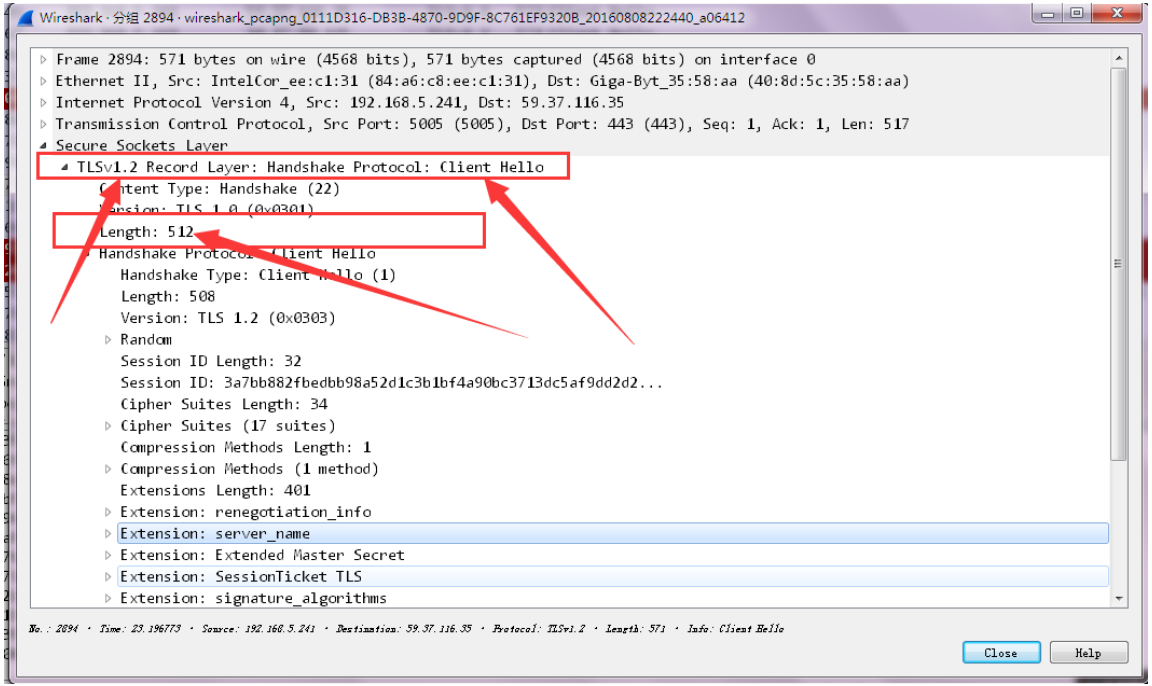
三 Wireshark分析SSL

- Wireshark（前称Ethereal)
  - 一个网络封包分析软件。
  - 网络封包分析软件的功能是撷取网络封包，并尽可能显示出最为详细的网络封包资料。
  - Wireshark使用WinPCAP作为接口，直接与网卡进行数据报文交换。本文着重分析wireshark捕获SSL协议及其对SSL协议的故障分析。



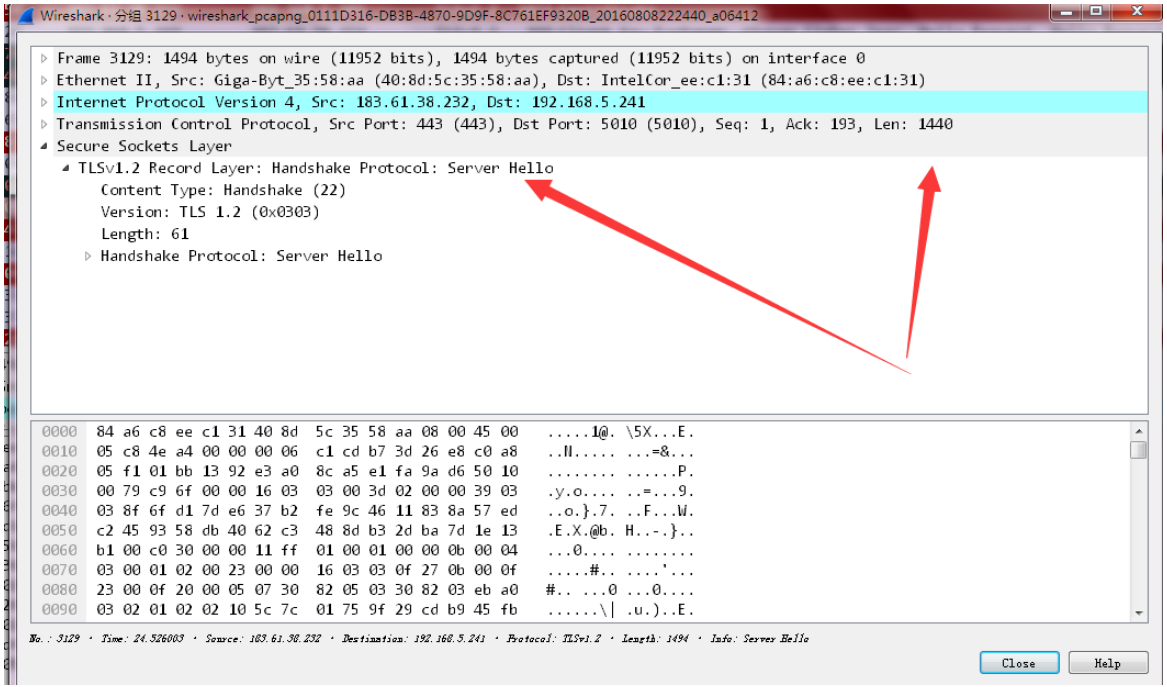
Wireshark分析SSL

- 客户端Hello包
  - 建立SSL协议前，客户端发送了请求包，进行握手协商



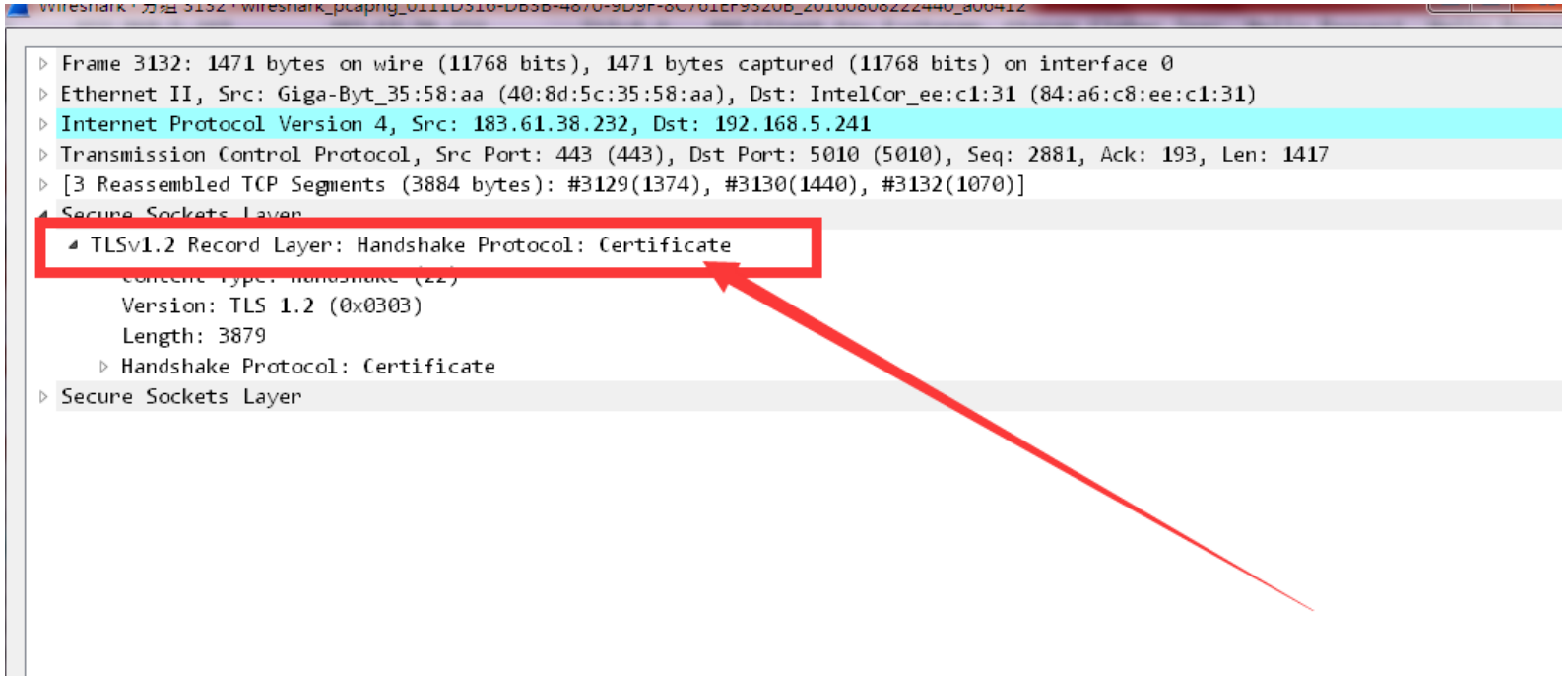
Wireshark分析SSL

- 服务器Hello包



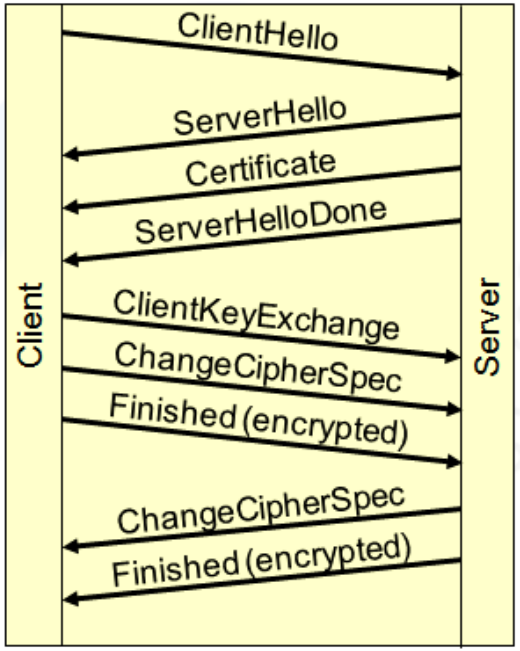
Wireshark分析SSL

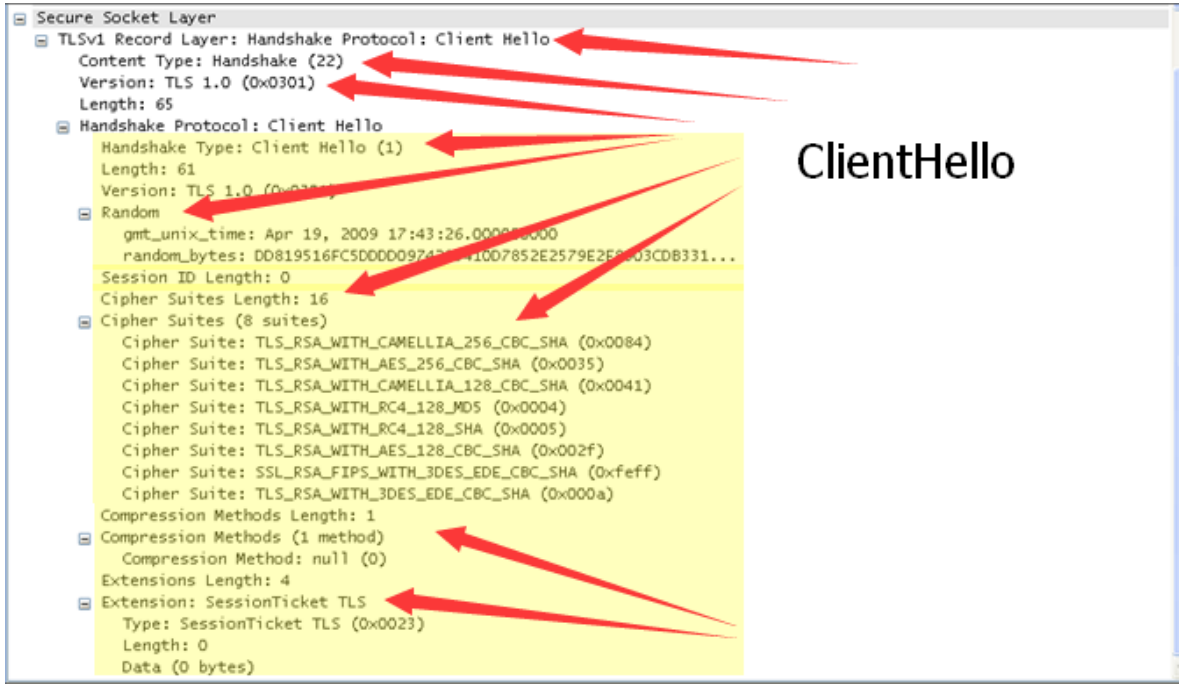
证书传输



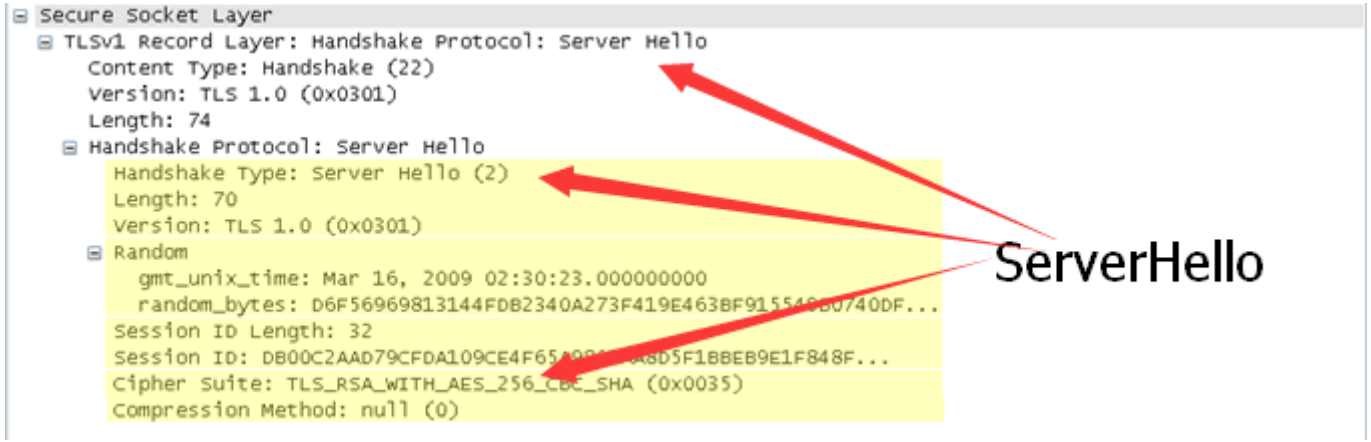
Wireshark分析SSL

常规的RSA握手



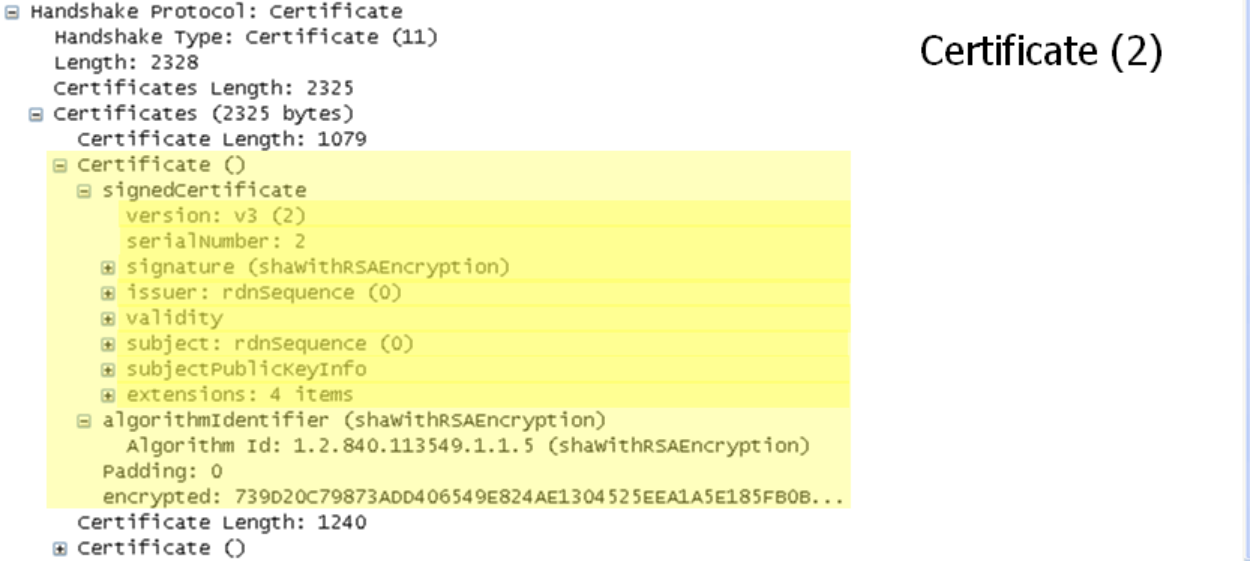


Wireshark分析SSL



Wireshark分析SSL

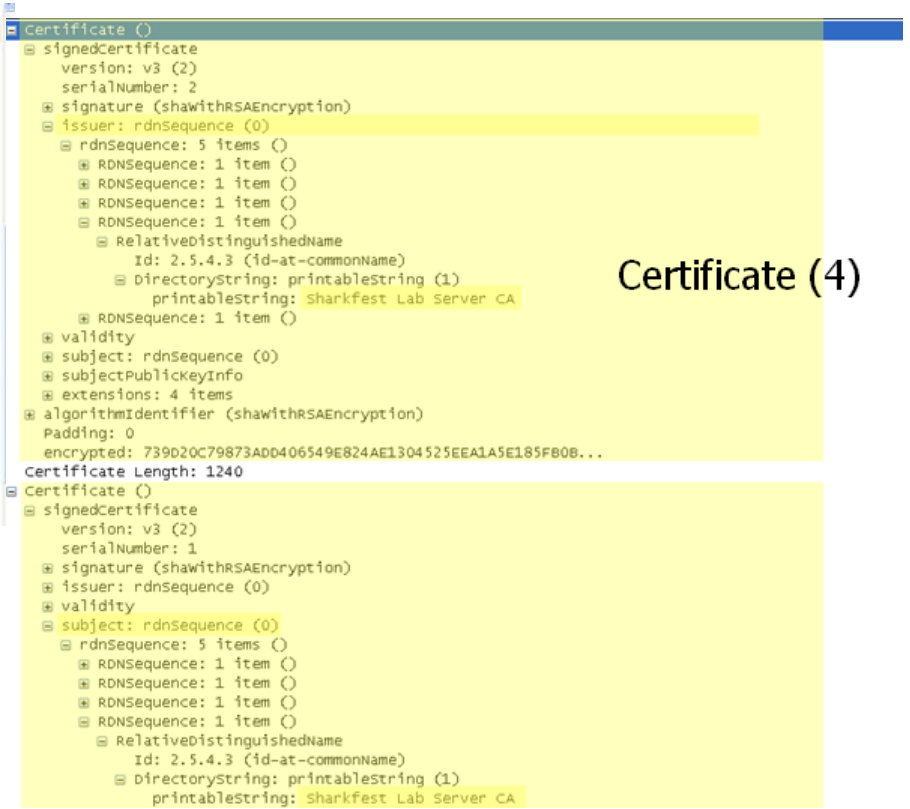
Certificate (1)



Certificate (2)

Wireshark分析SSL

Certificate (3)



Certificate (4)

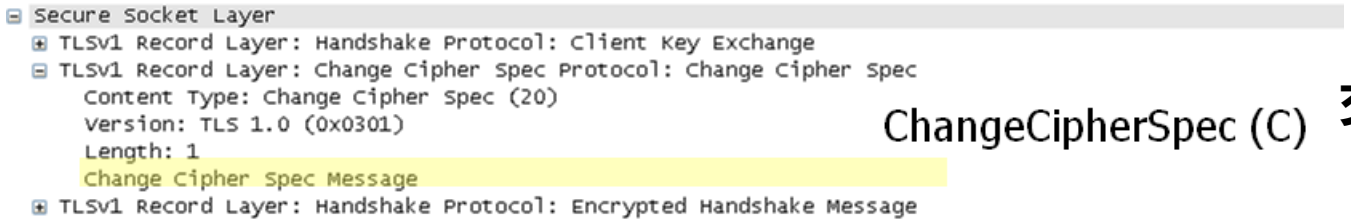
Wireshark分析SSL

ServerHelloDone

服务器证书加密

ClientKeyExchange

客户端密钥交换

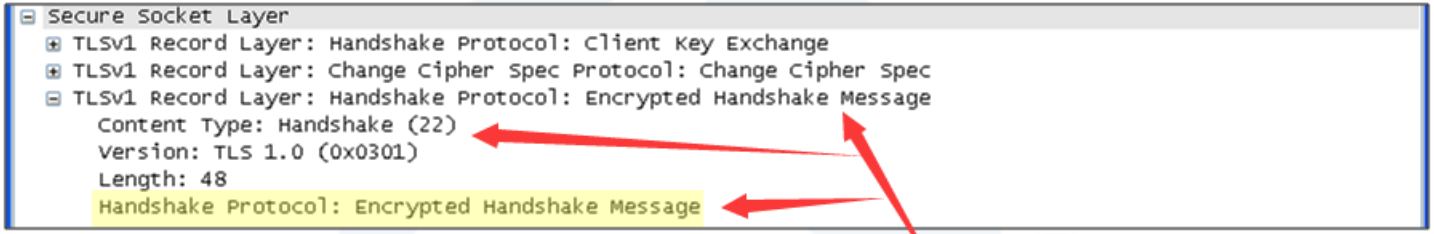


ChangeCipherSpec (C)

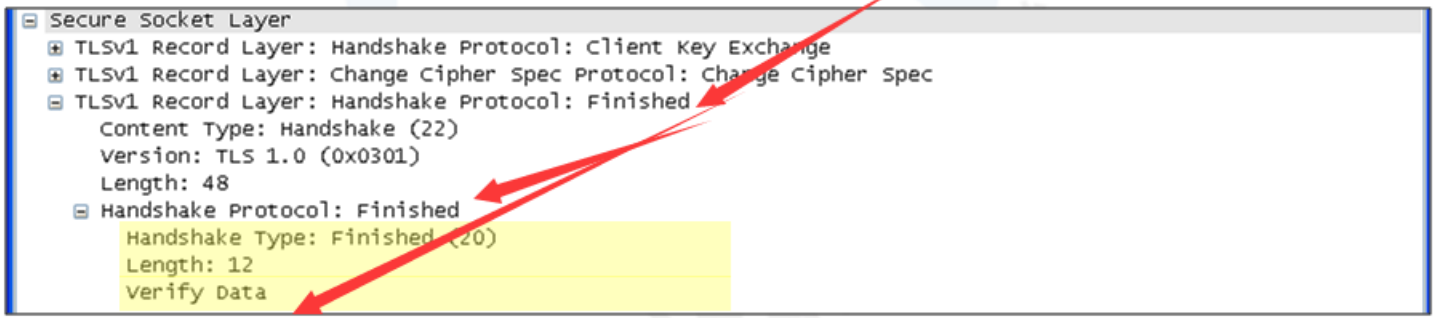
交换客户端加密套件

Wireshark分析SSL

Without decryption:



With decryption:



Finished (C)　　客户端完成加密套件加密

Wireshark分析SSL

```
Secure Socket Layer
  TLSv1 Record Layer: Change Cipher Spec Protocol: Change Cipher Spec
      Content Type: Change Cipher Spec (20)
      Version: TLS 1.0 (0x0301)
      Length: 1
      Change Cipher Spec Message
  TLSv1 Record Layer: Handshake Protocol: Encrypted Handshake Message
```

ChangeCipherSpec (S)

服务器交换加密套件

Without decryption:

```
Secure Socket Layer
  TLSv1 Record Layer: Change Cipher Spec Protocol: Change Cipher Spec
  TLSv1 Record Layer: Handshake Protocol: Encrypted Handshake Message
      Content Type: Handshake (22)
      Version: TLS 1.0 (0x0301)
      Length: 48
      Handshake Protocol: Encrypted Handshake Message
```

Finished (S)

服务器交换密钥套件完成

With decryption:

```
Secure Socket Layer
  TLSv1 Record Layer: Change Cipher Spec Protocol: Change Cipher Spec
  TLSv1 Record Layer: Handshake Protocol: Finished
      Content Type: Handshake (22)
      Version: TLS 1.0 (0x0301)
      Length: 48
    Handshake Protocol: Finished
        Handshake Type: Finished (20)
        Length: 12
        Verify Data
```

三 脆弱性分析

- 客户端假冒
  - 因为SSL协议设计初衷是对Web站点及网上交易进行安全性保护，使消费者明白正在和谁进行交易要比使商家知道谁正在付费更为重要，为了不致于由于安全协议的使用而导致网络性能大幅下降， **SSL协议并不是默认地要求进行客户认证**，这样做虽然有悖于安全策略，但却促进了SSL的广泛应用。
  - 针对这个问题，可在必要的时候配置SSL协议，使其选择对客户端进行认证鉴别。

三 脆弱性分析

- SSL协议无法提供基于UDP应用的安全保护
  - SSL协议需要在握手之前建立TCP连接，因此不能对UDP应用进行保护。如果要兼顾UDP协议层之上的安全保护，可以采用IP层的安全解决方案。

- SSL协议不能对抗通信流量分析
  - 由于SSL只对应用数据进行保护，数据包的IP头和TCP头仍然暴露在外，通过检查没有加密的IP源和目的地址以及TCP端口号或者检查通信数据量，一个通信分析者依然可以揭示哪一方在使用什么服务，有时甚至揭露商业或私人关系的秘密。

三 脆弱性分析

- 进程中主密钥泄漏
  - 除非SSL的工程实现大部分驻留在硬件中，否则主密钥将会存留在主机的主存储器中，这就意味着任何可以读取SSL进程存储空间的攻击者都能读取主密钥，因此，不可能面对掌握机器管理特权的攻击者而保护SSL连接，这个问题要依靠用户管理策略来解决。

- 磁盘上的临时文件可能遭受攻击
  - 对于使用虚拟内存的操作系统，某些敏感数据甚至主密钥都交换到磁盘上，可采取内存加锁和及时删除磁盘临时文件等措施来降低风险。

办公地点：理科大楼B1209

联系方式：17621203829

邮箱：liuhongler@foxmail.com