TLS/SSL study notes——komiko

What is TLS ？

▼ 在 OSI 参考模型中，系统之间的通信分为七个不同的抽象层

物理层（Physical Layer）

数据链路层（Data Link Layer）

网络层（Network Layer）

传输层（Transport Layer）

会话层（Session Layer）

表示层（Presentation Layer）

应用层（Application Layer）

TLS(Transport Layer Security)：

顾名思义，TSL协议是一种保证客户端-服务端传输安全的加密协议

- 目的：对 Web 应用和服务器之间的通信（例如，Web 浏览器加载网站）进行加密。

：TLS 还可以用于加密其他通信，如电子邮件、消息传递和 IP 语言（VOIP）等

最突出的用途是用于保护http协议(https),

- 作用：保证信息传递的**机密性**：加密消息

身份**验证**：确保交换信息的各方是他们所声称的身份

验证信息的**完整性**：验证数据未被伪造或篡改

TLS和SSL的区别：

SSL是TSL的直接前身，技术较老，包含一些安全漏洞，已经弃用，但是由于SSL知名度过大，有些时候SSL/TLS指的都是TLS

TLS 握手

TLS 连接建立的第一阶段是 TLS 握手协议。在握手过程中，通信双方交换信息以相互确定，彼此验证，其目标是在通信各方之间安全地协商共享对称密钥，实现通信信息的加密以及对信息来源的身份认证

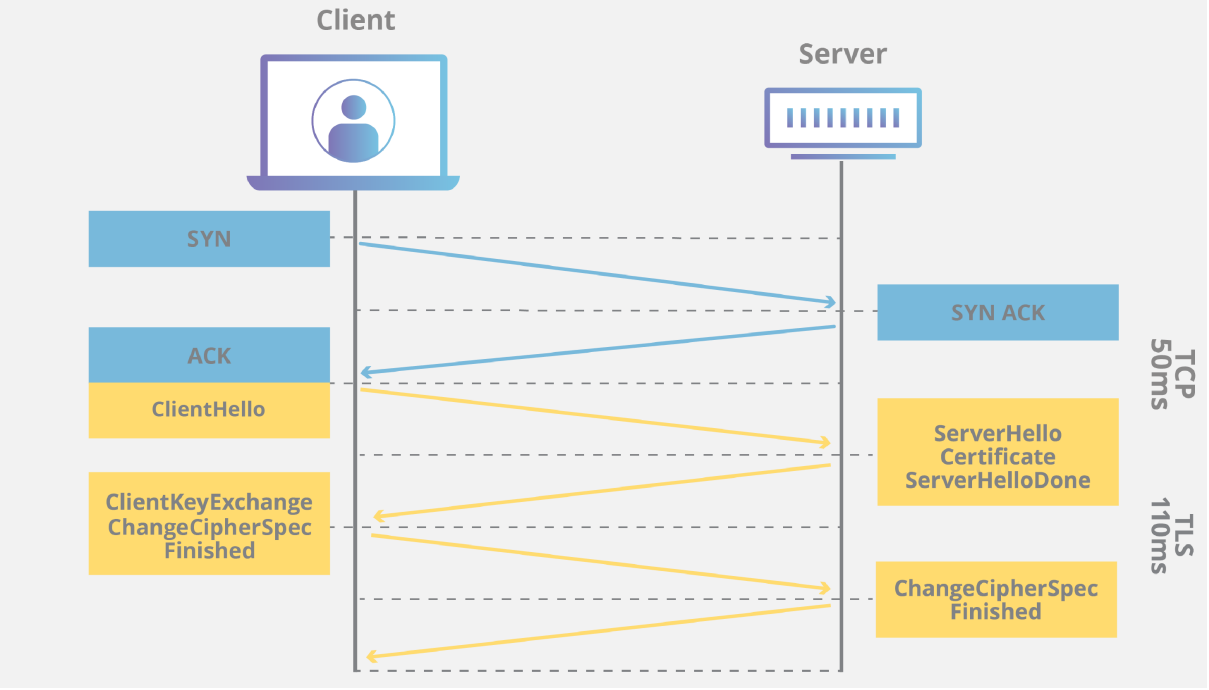
TSL握手有哪些步骤

TLS 握手中的确切步骤将根据所使用的密钥交换算法的种类和双方支持的密码套件而有所不同

* RSA密钥交换算法：

RSA原理详见：

RSA



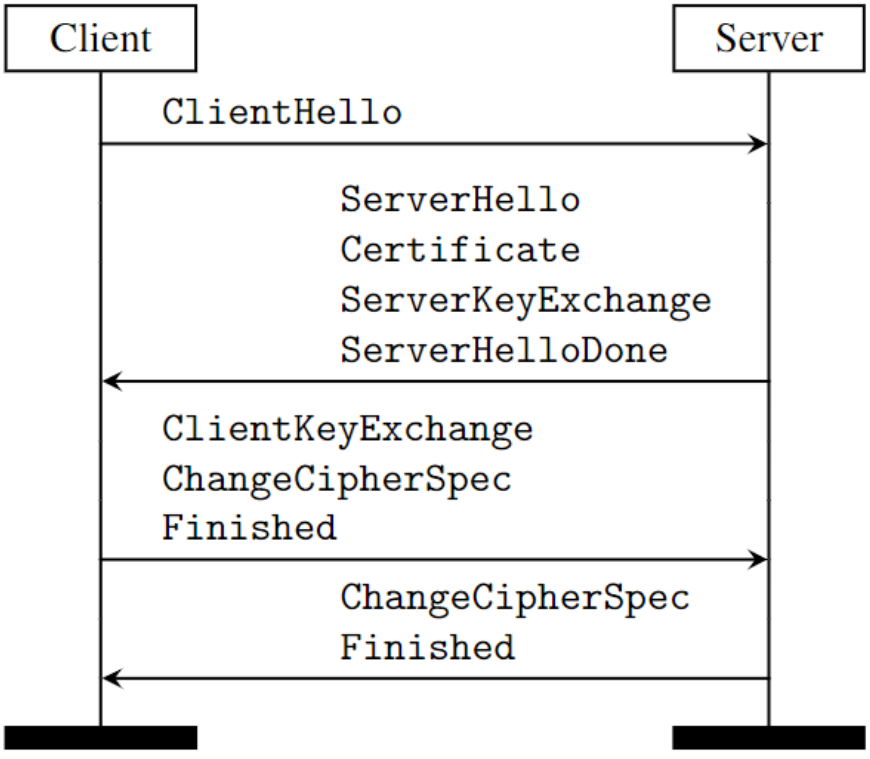
1. **client hello**：客户端向服务端发送问候消息开始握手，其中包括其支持哪种 TLS 协议和密码套件的信息
2. **server hello**：服务端对问候进行响应，包括其相应的TLS证书(certificate)，公钥交换(server key exchange),结束问候(server hello done)

3. **身份验证**： 客户端使用颁发该证书的证书颁发机构验证服务器的 SSL 证书。服务端确认身份
4. **预主密钥**：客户端发送随机预主密钥，预主密钥是使用公钥加密的，只能使用服务器的私钥解密。
5. **私钥被使用**：服务器对预主密钥进行解密
6. **生成会话密钥：**客户端和服务器均使用客户端随机数、服务器随机数和预主密钥生成会话密钥。双方应得到相同的结果。
7. **客户端就绪：**客户端发送一条“已完成”消息，该消息用会话密钥加密。
8. **服务器就绪：**服务器发送一条“已完成”消息，该消息用会话密钥加密。
9. **实现安全对称加密：**已完成握手，并使用会话密钥继续进行通信。

*DH协议密钥交换算法

DH原理详见：

DH密钥协议



1. **客户端问候**

2. **服务端问候**
3. **服务器的数字签名：**服务器对到此为止的所有消息计算出一个数字签名。
4. **数字签名确认：**客户端验证服务器的数字签名，确认服务器是它所声称的身份。
5. **客户端 DH 参数：**客户端将其 DH 参数发送到服务器。
6. **客户端和服务器计算预主密钥：**客户端和服务器使用交换的 DH 参数分别计算匹配的预主密钥，而不像 RSA 握手那样由客户端生成预主密钥并将其发送到服务器。
7. **创建会话密钥：**与 RSA 握手中一样，客户端和服务器现在从预主密钥、客户端随机数和服务器随机数计算会话密钥。
8. **客户端就绪：**与 RSA 握手相同。
9. **服务器就绪**
10. **实现安全对称加密**

TSL怎么提供完整性

SSL 和 TLS 通过计算消息摘要来提供数据完整性

Data integrity							
Algorithm	SSL 2.0	SSL 3.0	TLS 1.0	TLS 1.1	TLS 1.2	TLS 1.3	Status
HMAC-MD5	Yes	Yes	Yes	Yes	Yes	No	Defined for TLS 1.2 in RFCs
HMAC-SHA1	No	Yes	Yes	Yes	Yes	No	
HMAC-SHA256/384	No	No	No	No	Yes	No	
AEAD	No	No	No	No	Yes	Yes	
GOST 28147-89 IMIT[79]	No	No	No	No	Yes	No	Defined for TLS 1.2 in RFC 9189.
GOST R 34.12-2015 AEAD[79]	No	No	No	No	No	Yes	Defined for TLS 1.3 in RFC 9367.

利用哈希算法的不可逆性，检验数据传输前后是否一致（完整性）