TLS 握手期间会发生什么？| SSL 握手

在 TLS/SSL 握手过程中，客户端和服务器交换 SSL 证书、密码套件要求以及为创建会话密钥而随机生成的数据。

学习目标

阅读本文后，您将能够：

- 了解什么是 TLS 握手
- 了解 TLS 握手的目的
- 说明 TLS 握手的步骤
- 探索不同类型的 TLS 握手

相关内容

什么是 SSL 证书?

Keyless SSL

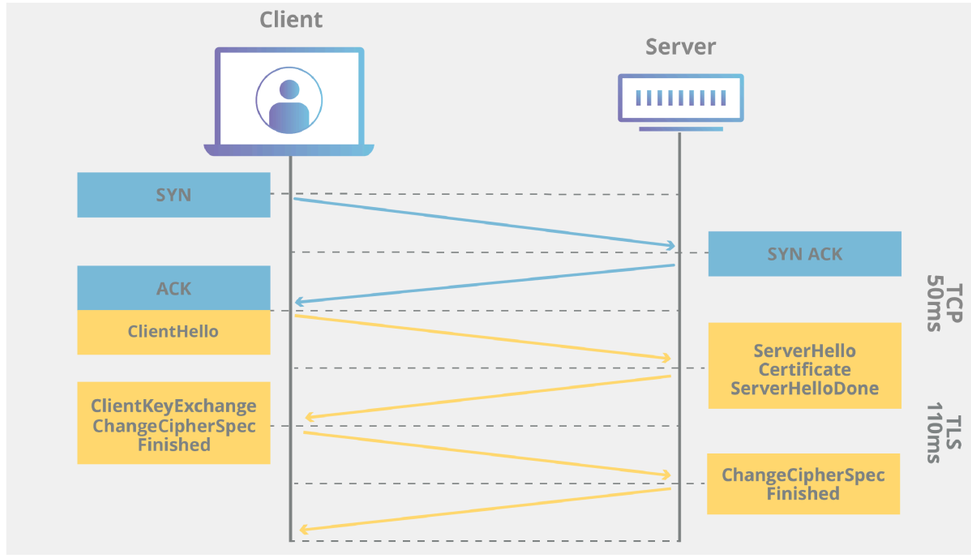
什么是 SSL?

什么是混合内容?

什么是 HTTPS?

复制文章链接

什么是 TLS 握手?



TLS 是一种旨在保护互联网通信安全的加密和身份验证协议。TLS 握手是启动 TLS 通信会话的过程。在 TLS 握手过程中，通信双方交换消息以相互确认，彼此验证，确立它们将使用的加密算法，并生成一致的会话密钥。TLS 握手是 HTTPS 工作原理的基础部分。

TLS 与 SSL 握手

SSL（安全套接字层）是为 HTTP 开发的原始安全协议。不久前，SSL 被 TLS（Transport Layer Security）所取代。SSL 握手现在称为 TLS 握手，尽管“SSL”这个名称仍在广泛使用。

何时进行 TLS 握手?

用户导航到一个使用 HTTPS 的网站，浏览器首先开始查询网站的源服务器，这时就会发生 TLS 握手。在任何其他通信使用 HTTPS 时（包括 API 调用和 DNS over HTTPS 查询），也会发生 TLS 握手。

通过 TCP 握手打开 TCP 连接后，将发生 TLS 握手。

TLS 握手期间会发生什么?

在 TLS 握手过程中，客户端和服务器一同执行以下操作：

- 指定将要使用的 TLS 版本（TLS 1.0、1.2、1.3 等）
- 决定将要使用哪些密码套件（见下文）
- 通过服务器的公钥和 SSL 证书颁发机构的数字签名来验证服务器的身份

TLS 握手有哪些步骤?

TLS 握手是由客户端和服务器交换的一系列数据报或消息。TLS 握手涉及多个步骤，因为客户端和服务器要交换完成握手和进行进一步对话所需的信息。

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 密钥交换算法虽然现在被认为不安全，但曾在 1.3 之前的 TLS 版本中使用。大致如下：

1. **“客户端问候（client hello）”消息：** 客户端通过向服务器发送“问候”消息来开始握手。该消息将包含客户端支持的 TLS 版本，支持的密码套件，以及称为一串称为“客户端随机数（client random）”的随机字节。

2. **“服务器问候（server hello）”消息：** 作为对 client hello 消息的回复，服务器发送一条消息，内含服务器的 SSL 证书、服务器选择的密码套件，以及“服务器随机数（server random）”，即由服务器生成的另一串随机字节。

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。此举确认服务器是其声称的身份，且客户端正在与该域的实际所有者进行交互。

4. **预主密钥：** 客户端再发送一串随机字节，即“预主密钥（premaster secret）”。预主密钥是使用公钥加密的，只能使用服务器的私钥解密。（客户端从服务器的 SSL 证书中获得公钥。）

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。

所有 TLS 握手均使用非对称加密（公钥和私钥），但并非全都会在生成会话密钥的过程中使用私钥。例如，短暂的 Diffie-Hellman 握手过程如下：

1. **客户端问候：**客户端发送客户端问候消息，内含协议版本、客户端随机数和密码套件列表。

2. **服务器问候：**服务器以其 SSL 证书、其选定的密码套件和服务器随机数回复。与上述 RSA 握手相比，服务器在此消息中还包括以下内容（步骤 3）：

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**

*DH 参数：DH 代表 Diffie-Hellman。Diffie-Hellman 算法使用指数计算得出相同的预主机密。服务器和客户端各自提供用于计算的参数，并且组合后在每一端产生不同的计算，但得出相等的结果。

要详细了解临时 Diffie-Hellman 握手与其他类型握手之间的区别，以及它们如何实现前向保密，请参阅什么是 Keyless SSL?

TLS 1.3 中的握手有什么不同?

TLS 1.3 不支持 RSA，也不支持易受攻击的其他密码套件和参数。它还缩短了 TLS 握手，使 TLS 1.3 握手更快更安全。

TLS 1.3 握手的基本步骤为：

- **客户端问候：**客户端发送客户端问候消息，内含协议版本、客户端随机数和密码套件列表。由于已从 TLS 1.3 中删除了对不安全密码套件的支持，因此可能的密码套件数量大大减少。客户端问候消息还包括将用于计算预主密钥的参数。大体上来说，客户端假设它知道服务器的首选密钥交换方法（由于简化的密码套件列表，它有可能知道）。这减少了握手的总长度——这是 TLS 1.3 握手与 TLS 1.0、1.1 和 1.2 握手之间的重要区别之一。

- **服务器生成主密钥：**此时，服务器已经接收到客户端随机数以及客户端的参数和密码套件。它已经拥有服务器随机数，因为它可以自己生成。因此，服务器可以创建主密钥。

- **服务器问候和“完成”：**服务器问候包括服务器的证书、数字签名、服务器随机数和选择的密码套件。因为它已经有了主密钥，所以它也发送了一个“完成”消息。

- **最后步骤和客户端“完成”：**客户端验证签名和证书，生成主密钥，并发送“完成”消息。

- **实现安全对称加密**

0-RTT 模式用于会话恢复

TLS 1.3 还支持一个更快的 TLS 握手版本，根本无需往返，或客户端与服务器之间来回通信。如果客户端与服务器之前已经相互连接（例如，如果用户之前访问过该网站），它们可以各自从第一个会话中获取另一个共享密钥，称为“恢复主密钥”。在第一个会话期间，服务器还会向客户端发送称为会话票证的内容。客户端可以使用此共享密钥，在下一次会话的第一条消息中将加密数据连同该会话票证一起发送到服务器。然后客户端与服务器之间的 TLS 恢复。

什么是密码套件？

密码套件是一组用于建立安全通信连接的算法。广泛使用的密码套件有多种，而且 TLS 握手的一个重要组成部分就是针对这一握手使用哪个密码套件达成一致意见。

要进一步了解 TLS/SSL，请参阅 SSL 如何工作？ 。