

# TLS1.2握手流程

TLS1.2 的握手过程包括以下七个阶段：

1. 客户端发送 Client Hello 消息，包含客户端支持的加密算法、应用协议等信息。
2. 服务端收到 Client Hello 消息后，发送 Server Hello 消息，包含选定的加密算法、服务器端证书等信息。
3. 服务端发送 Certificate、Server Key Exchange、Certificate Request 三个消息，用于双向认证、协商会话密钥和请求客户端证书。
4. 客户端接收到服务器端的消息后，决定是否发送自己的证书，如果需要则发送 Certificate 消息。
5. 客户端发送 Client Key Exchange 消息，用于协商会话密钥。
6. 客户端发送 Change Cipher Spec 消息，告知服务器端后续传输的数据将使用协商好的加密算法进行加密。
7. 客户端发送 Finished 消息，服务器端接收到 Finished 消息后，也发送 Change Cipher Spec 和 Finished 消息，表示握手完成。

关于校验证书/单向认证/双向认证三种方式在握手流程上的区别如下：

## 1. 不校验证书

在不校验证书的情况下，客户端和服务端之间的握手过程与通常的握手过程类似。但是，在服务器端不发送证书或者客户端忽略证书的情况下，客户端无法验证服务器端的身份，存在被中间人攻击篡改数据的风险。

## 1. 单向认证

在单向认证中，服务器端发送自己的证书给客户端，在握手过程中客户端验证服务器端证书的有效性。客户端可以不发送自己的证书，此时服务器端无法验证客户端的身份。在单向认证的情况下，客户端可以确保服务器端的身份是可信的，但服务器端无法验证客户端的身份，存在数据被窃听和伪造等风险。

## 1. 双向认证

在双向认证中，客户端和服务端都需要发送自己的证书，并验证对方证书的有效性。客户端和服务端都可以确认对方的身份，相互信任后进入加密通信状态。在双向认证的情况下，安全性更高，但是握手过程相对来说比较复杂，且需要在客户端和服务端都配置和管理证书。

## 1.如果要完成双向认证，该如何部署证书呢？

首先，双向认证，即客户端需要验证服务端证书，服务端也需要验证客户端证书。

举个例子：

现在我自己生成了一个自签名证书，并且用它签发了服务端证书server.crt和客户端证书client.crt。那么

对于客户端来说：**服务端需要加载CA证书和服务端证书。CA证书用于验证客户端证书的合法性，服务端证书用于向客户端证明服务器的身份。**

对于服务端来说：**客户端需要加载CA证书和客户端证书。CA证书用于验证服务器证书的合法性，客户端证书用于向服务器证明客户端的身份。**

以下是详细步骤。

## 1. 在服务端部署证书

服务端需要加载CA证书和服务器证书。CA证书用于验证客户端证书的合法性，服务器证书用于向客户端证明服务器的身份。

具体步骤如下：

- 将CA证书和服务器证书保存到服务端的某个目录下。
- 在服务端的配置文件中配置证书路径和密码等信息。

1. 在客户端部署证书

客户端需要加载CA证书和客户端证书。CA证书用于验证服务器证书的合法性，客户端证书用于向服务器证明客户端的身份。

具体步骤如下：

- 将CA证书和客户端证书保存到客户端的某个目录下。
- 在客户端的配置文件中配置证书路径和密码等信息。

需要注意的是，证书的部署和配置可能因具体的应用场景而有所不同，以上仅为一般情况的示例。

疑问：为什么我没有安装CA证书，连接公司的域名也可以成功呢？

这可能是因为公司的域名使用了公共CA机构颁发的证书，而这些证书已经被大多数操作系统和浏览器所信任。因此，即使您的客户端没有安装该公司的CA证书，也可以成功验证对端证书。但是，如果该公司使用自签名证书或私有CA机构颁发的证书，那么客户端需要安装该公司的CA证书才能成功验证对端证书。

## 2.关于证书链

2023/5/26 11:47:39

我举一个例子：我有一个ca证书叫做sakura\_A，我用它签发了证书sakura\_AA，然后用sakura\_AA签发服务端证书sakura\_AAA和sakura\_AAB，并将sakura\_AAA和sakura\_BBB部署在对应的机器上，那么如果我要进行双向认证，需要用sakura\_A来验证对端证书还是用sakura\_AA来验证对端证书呢？

2023/5/26 11:48:20

在您的示例中，sakura\_A是您的根证书，sakura\_AA、sakura\_AAA和sakura\_AAB都是由它签发的下级证书。在进行双向认证时，客户端需要验证服务器的数字证书，服务器也需要验证客户端的数字证书。

对于服务器sakura\_AAA和sakura\_AAB，客户端需要使用sakura\_A或sakura\_AA来验证其数字证书的可信性。因为这两个证书是由根证书sakura\_A签发的，并且拥有自身的数字签名，因此只要根证书sakura\_A或签发它的下级证书sakura\_AA验证通过，就可以确认其可信。

对于客户端证书，由于客户端提供的证书不是由根证书签发，而是由根证书签发的下级证书sakura\_AA签发的，因此服务器需要使用根证书sakura\_A或签发它的下级证书sakura\_AA来验证客户端证书的可信性。

所以，双向认证中客户端需要验证服务器证书的可信性，使用sakura\_A或sakura\_AA来验证；服务器需要验证客户端证书的可信性，使用sakura\_A或sakura\_AA来验证。

## 3.双向认证的流程中时间起了什么作用？

在双向认证的流程中，时间起到了至关重要的作用。数字证书中包含有证书的有效期，双向认证的过程需要校验数字证书的有效性。其原理是客户端和服务端分别比对数字证书的签名和摘要信息，来判断数字证书是否真实、有效和未过期。

同时，在双向认证过程中，加密算法（如RSA、DSA等）也需要使用当前时间作为种子生成随机数或伪随机数，用于生成密钥和签名。此外，还有很多需要精确时间控制的操作，如计算证书链的哈希值、计算通信加密算法的密钥和IV值等等。

如果系统时间不准确或者与服务端时间不一致，可能会导致双向认证失败或者安全性降低，甚至可能引发重大安全风险。比如，如果数字证书已经过期，客户端就不能够和服务端建立安全的通信连接；而如果系统时间被篡改，攻击者就有可能通过替换数字证书以及伪造签名和摘要信息，从而欺骗认证服务器端，进而获取重要的敏感数据。

因此，在数字证书的生成、存储、传输和使用过程中，需要确保系统时间的准确性和一致性，以保证双向认证的安全性和可靠性。可以通过定时同步网络时间协议（NTP）或者使用时钟同步程序等方式来确保系统时间的准确性。

## 4.双向认证过程中，双端是怎么确认证书有效期的？系统时间会有影响吗？

简单理解就是一句话：端上需要使用系统时间来验证对端证书的有效期。

在双向认证过程中，双方需要验证对方的数字证书是否有效。数字证书中包含了证书持有者的公钥、个人信息和数字签名等信息，其中就包括证书的有效期。客户端和服务端分别比对数字证书的签名和摘要信息，来判断数字证书是否真实、有效和未过期。

具体来说，双向认证流程中，当客户端向服务端发送连接请求时，服务器会将数字证书返回给客户端，然后客户端会验证数字证书的签名和摘要信息，以确保数字证书的真实性和合法性。其中，验证证书有效期的方法是比对当前系统时间（也就是客户端本地时间）和数字证书中记录的有效期限范围是否一致。如果数字证书已经过期，或者证书生效时间还没有到达，那么客户端就会拒绝与服务端建立安全通信连接。

因此，双向认证确实依赖于系统时间的准确性和一致性，系统时间的误差可能会导致认证失败或者安全性降低。如果系统时间不准确或者与服务端时间不一致，就有可能导致数字证书的有效期被误判，从而引发安全问题。为了避免这种情况的发生，有必要确保系统时间的准确性和一致性，建议使用网络时间协议（NTP）或其他可靠的时钟同步程序，来同步客户端和服务端的当前系统时间。

补充说明，认证过程中，系统时间必须要在证书有效期内，服务端证书、根证书、客户端证书的有效期取交集。也就是说，系统时间会卡三个证书的有效期起止时间，在范围内才可以完成认证。

在双向认证的情况下，客户端和服务端都需要拥有有效的证书。每个证书都有自己的有效期，包括起始日期和截止日期。

对于双向认证来说，客户端和服务端的证书有效期并没有强制要求保持一致。每个证书的有效期是独立验证的，客户端和服务端会分别检查对方的证书有效期。

系统时间对于证书的验证是非常重要的。客户端和服务端都会使用当前的系统时间与证书的有效期进行比较。如果系统时间不准确，可能会导致证书验证失败。

为了确保双向认证的成功，**系统时间应该与证书的有效期限有重叠的区间。即系统时间应该早于证书的截止日期，并且晚于证书的起始日期。只有在这个区间内，证书的有效期验证才会通过。**