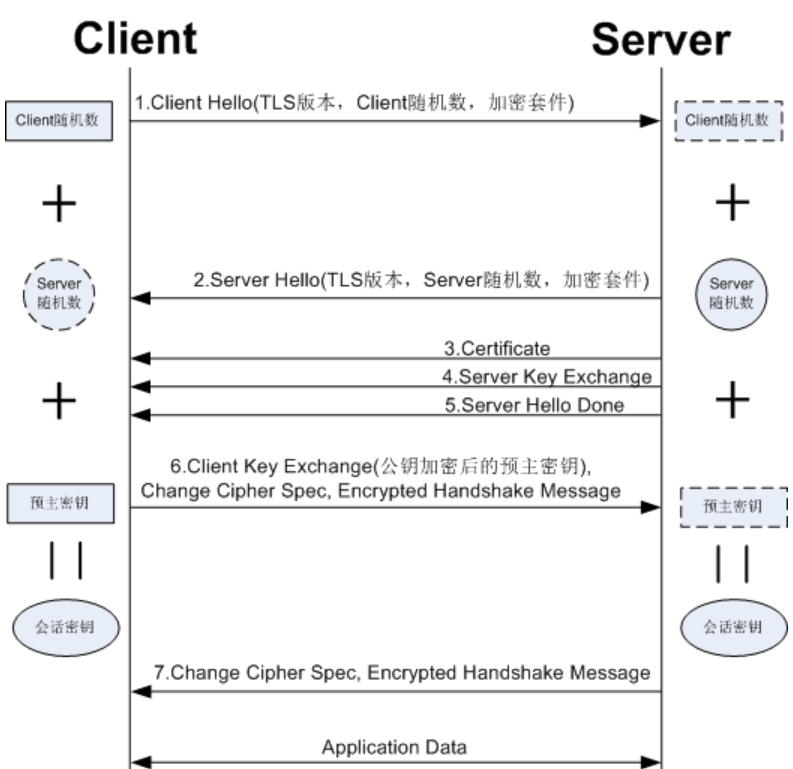
[https://ssl-config.mozilla.org/#server=nginx&version=1.22.0&config=intermediate&openssl=1.1.1q&hsts=false&ocsp=false&guideline=5.6](https://ssl-config.mozilla.org/" \l "server=nginx&version=1.22.0&config=intermediate&openssl=1.1.1q&hsts=false&ocsp=false&guideline=5.6" \t "_blank)

这个网站是关于TLS配置文件的生成器，根据不同的服务器类型、软件版本和安全策略，生成最佳的 TLS 配置。

<https://ciphersuite.info/cs/>

这个网站详细列出了不同的加密套件cipher suite提供有关它们的详细信息，包括支持的加密算法、密钥交换方法、消息认证码 (MAC) 算法等，例如在实际的项目生产环境中选择合适的加密套件

TLS1.2握手流程图



**1.Client Hello(TLS 版本，加密套件，Client 随机数)**

Client 首先发送 Hello，告诉 Server 自己所支持的 TLS 版本以及支持的加密套件，再把 Client 随机数发给 Server。

**2.Server Hello**

Server 收到 Client Hello 也给 Client 发送 Hello，并告知 Server 确认支持的 TLS 版本以及选择的加密算法。同样，也会把 Server 随机数发给 Client。

**3.Certificate**

Server 接着再发送自己的证书给 Client，这样 Client 浏览器就可以对照自己的证书信任列表来确认这个服务器是否可信。

**4.Server Key Exchange**

发送完证书，Server 就把公钥发送给 Client。

这个公钥是在密钥交换阶段使用的，用于客户端和服务器之间建立一个共享的密钥。服务器会提供一个临时公钥，客户端使用它来生成一个共享的秘密密钥（用于后续加密数据的会话密钥，也就是对称密钥）。

**5.Server Hello Done**

最后，Server 告诉 Client 响应发送完了。

**6.Client Key Exchange, Change Cipher Spec, Encrypted Handshake Message**

Client 用刚刚收到的公钥加密 pre-master 预主密钥，并把这个加密后的随机数发送给 Server。

Change Cipher Spec: 这一步就是告诉服务器往后的数据就用商议好的算法和密钥来加密。

Encrypted Handshake Message: 表示 Client 这边的 TLS 协商已经 OK 了。

Server 收到加密后的预主密钥，会用自己的私钥进行解密从而得到预主密钥。

Client 会用 Client 随机数、Server 随机数和预主密钥计算出会话密钥。Server 也同样计算出和 Client 一样的会话密钥。之后，Client 和 Server 就都只使用这个会话密钥(对称密钥)对数据进行加密。

**7.Change Cipher Spec, Encrypted Handshake Message**

这一步表示 Server 这边 TLS 协商也 OK 了，可以给数据进行加密交换了。