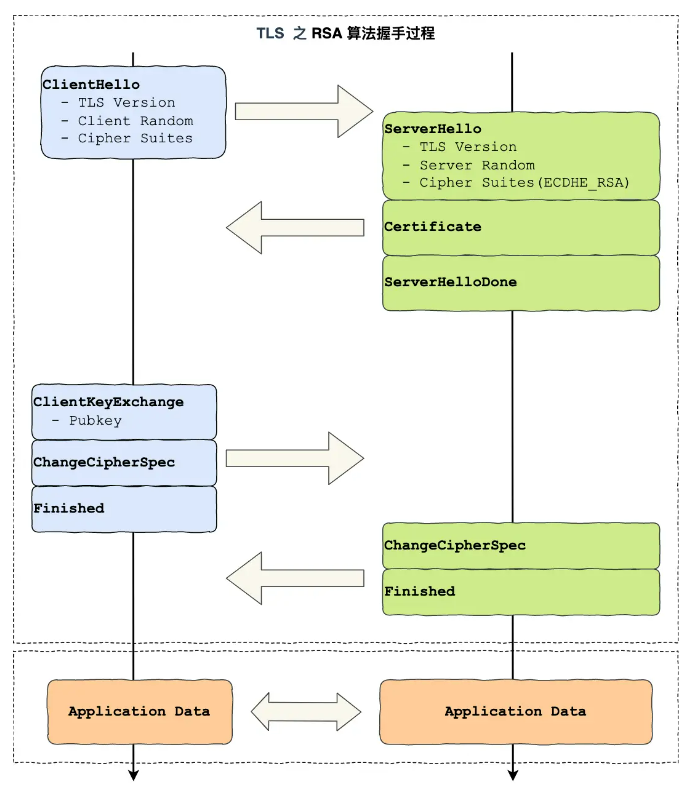
HTTPS RSA握手解析

1. TLS握手过程



每个框都是一个记录（TLS收发基本单位），多个记录可以组成一个TCP包发送，经过四个消息就可以完成TLS握手，需要两个RTT时延

不同的密钥交换算法TLS握手过程有区别

RSA握手过程：传统TLS握手基本使用RSA算法实现密钥交换，RSA密钥协商算法中客户端生成随机密钥，使用服务端公钥加密后再传给服务端，服务端通过私钥解密后得到相同密钥，再用它加密应用消息。

TLS第一次握手：客户端首先发送一个Client Hello消息（TLS版本，随机数，支持的密码套件列表）

TLS第二次握手：服务端收到客户端的Client Hello消息后确认TLS版本是否支持，从密码套件列表选择一个密码套件，生成随机数。**记录1**：返回Sever Hello消息（确认的TLS版本号，随机数，选择的密码套件（密码套件的格式为：密钥交换算法+签名算法+对称加密算法+摘要算法））。**记录2**：发送Sever Certificate给客户端，这个消息里有数字证书。**记录3**：发送Server Hello Done消息，表示本次打招呼结束。

TLS第三次握手：客户端验证完证书（证书内容（公钥，持有者信息，证书认证机构的信息，CA对这份文件的数字签名及使用的算法，证书有效期，额外的信息），证书签发（信息打包，计算hash值，CA私钥对hash值加密），客户端验证证书（客户端计算hash值，用CA公钥解密CA私钥加密的hash值，比较两个值是否一致），证书信任链）认为可信则继续。客户端生成一个新的随机数（pre-master key），用服务器公钥加密pre-master key，通过**记录1**：Client Key Exchange消息传给服务端，服务端收到后用私钥解密，得到pre-master key，至此客户端和服务端共享了三个随机数（Client Random， Server Random，pre-master key），根据这些随机数生成会话密钥（对称密钥，用于后续的HTTP请求/响应数据的加密）。生成对称密钥后客户端发送一个**记录2**：Change Cipher Spec，告诉服务端开始使用加密方式发送消息。客户端再发送一个**记录3**：Encrypted Handshake Message（Finished）消息，把之间发送的数据做个摘要用会话密钥加密让服务器验证加密通信是否可用及之前的握手信息是否被篡改。Change Cipher Spec之前都是明文传输，之后加密

TLS第四次握手：服务器发送**记录1**：Change Cipher Spec**记录2**：Encrypted Handshake Message，握手正式完成

RSA算法的缺陷：不支持前向保密（前向安全：拿到当前密钥也无法恢复以前的密钥），服务端私钥泄露过去被第三方截获的所有TLS通讯密文都会被破解