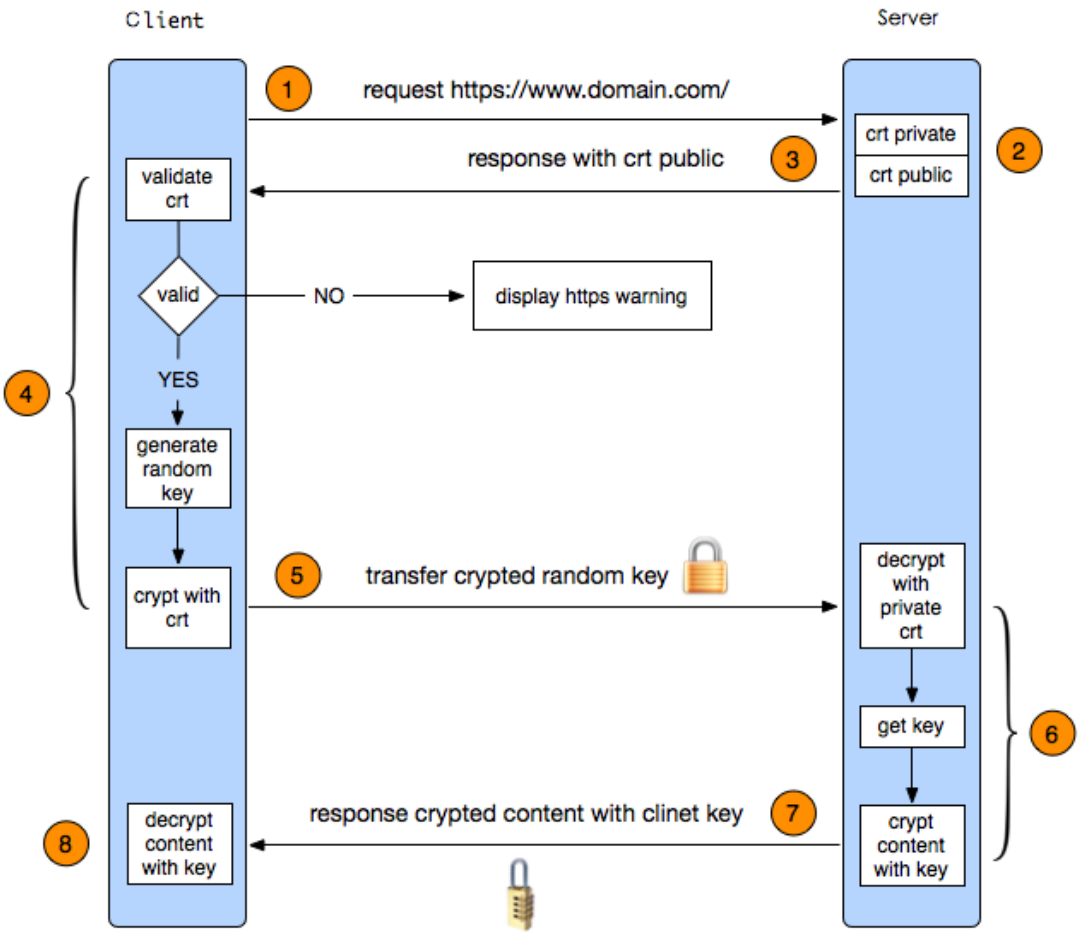
HTTPS 介绍

HTTPS 简介

HTTPS 其实是有两部分组成：HTTP + SSL / TLS，也就是在 HTTP 上又加了一层处理加密信息的模块。服务端和客户端的信息传输都会通过 TLS 进行加密，所以传输的数据都是加密后的数据。具体是如何进行加密，解密，验证的，且看下图。



1. 客户端发起 HTTPS 请求

这个没什么好说的，就是用户在浏览器里输入一个 https 网址，然后连接到 server 的 443 端口。

2. 服务端的配置

采用 HTTPS 协议的服务器必须要有一套数字证书，可以自己制作，也可以向组织申请。区别就是自己颁发的证书需要客户端验证通过，才可以继续访问，而使用受信任的公司申请的证书则不会弹出提示页面(startssl 就是个不错的选择，有 1 年的免费服务)。这套证书其

实就是一对公钥和私钥。如果对公钥和私钥不太理解，可以想象成一把钥匙和一个锁头，只是全世界只有你一个人有这把钥匙，你可以把锁头给别人，别人可以用这个锁把重要的东西锁起来，然后发给你，因为只有你一个人有这把钥匙，所以只有你才能看到被这把锁锁起来的东西。

3. 传送证书

这个证书其实就是公钥，只是包含了很多信息，如证书的颁发机构，过期时间等等。

4. 客户端解析证书

这部分工作是有客户端的 TLS 来完成的，首先会验证公钥是否有效，比如颁发机构，过期时间等等，如果发现异常，则会弹出一个警告框，提示证书存在问题。如果证书没有问题，那么就生成一个随即值。然后用证书对该随机值进行加密。就好像上面说的，把随机值用锁头锁起来，这样除非有钥匙，不然看不到被锁住的内容。

5. 传送加密信息

这部分传送的是用证书加密后的随机值，目的就是让服务端得到这个随机值，以后客户端和服务端的通信就可以通过这个随机值来进行加密解密了。

6. 服务段解密信息

服务端用私钥解密后，得到了客户端传过来的随机值(私钥)，然后把内容通过该值进行对称加密。所谓对称加密就是，将信息和私钥通过某种算法混合在一起，这样除非知道私钥，不然无法获取内容，而正好客户端和服务端都知道这个私钥，所以只要加密算法够彪悍，私钥够复杂，数据就够安全。

7. 传输加密后的信息

这部分信息是服务段用私钥加密后的信息，可以在客户端被还原

8. 客户端解密信息

客户端用之前生成的私钥解密服务段传过来的信息，于是获取了解密后的内容。整个过程第三方即使监听到了数据，也束手无策。

一次完整的 HTTPS 的通信步骤

步骤 1：客户端向服务器发送 ClientHello 报文，请求建立 SSL 连接。

ClientHello 报文：

客户端支持的 SSL 版本
客户端支持的加密算法
客户端支持的密钥长度

步骤 2：服务器收到客户端的请求后，向客户端发送 ServerHello 报文。

ServerHello 报文：

决定使用的 SSL 版本

决定使用的加密算法

步骤 3：服务器继续发送 Certificate 报文，即服务器的数字证书，其中包含服务器的公开密钥。

步骤 4：服务器发送 ServerHelloDone 报文，通知客户端进最初阶段的 SSL 握手协商部分结束。

步骤 5：客户端收到以上所有信息后，发送 ClientKeyExchange 报文作为回应。该报文已使用步骤 3 中的公开密钥加密。其中包含一种称为 Pre-master secret 的随机密码串，用于之后的对称秘钥加密通信。

步骤 6：客户端继续发送 ChangeCipherSpec 报文，该报文告诉服务器，在此之后的通信都会采用步骤 5 中的 Pre-master secret 秘钥加密。

步骤 7：客户端发送 Finished 报文。改报文包含连接至今全部报文的整体校验值。

步骤 8：服务端对客户端报文校验后，同样发送 ChangeCipherSpec 报文，含义与步骤 6 中的相同。

步骤 9：服务端发送 Finished 报文。

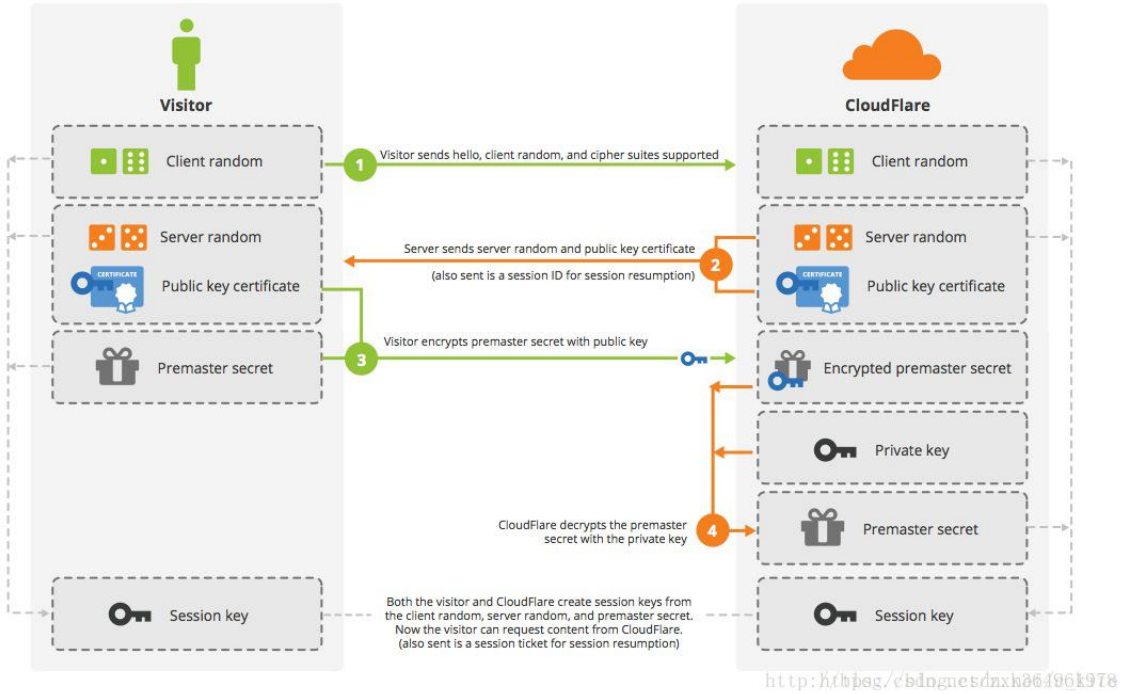
步骤 10：服务端和客户端的 Finished 报文交换完毕后，SSL 连接建立完成。从此后开始进行应用层协议的通信，即 HTTP 通信。

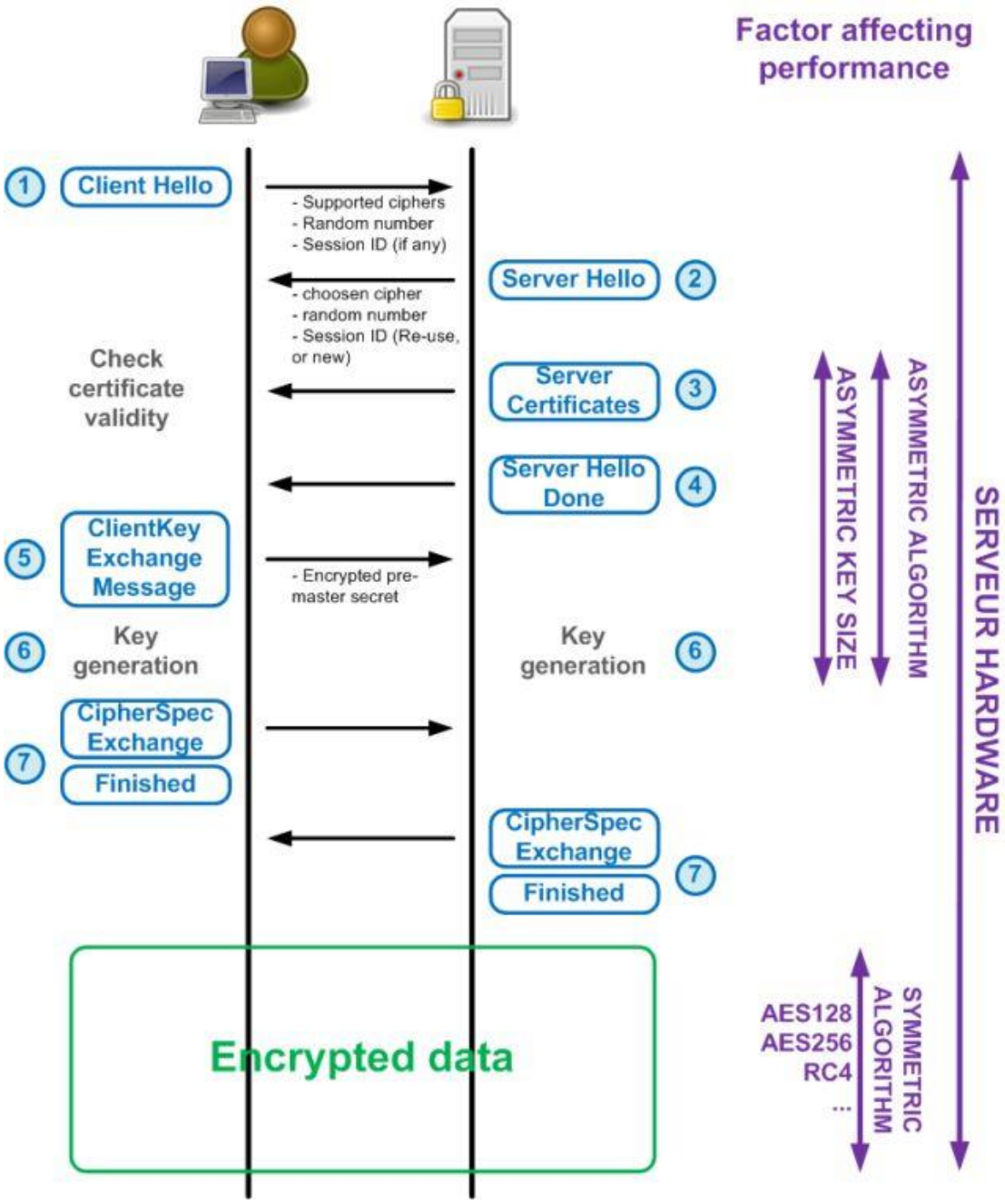
步骤 11：HTTP 通信。

步骤 12：客户端发送 close_notify 报文请求断开连接。之后再发送 TCP FIN 报文来关闭 TCP 通信。

SSL Handshake (RSA) Without Keyless SSL

Handshake




Factor affecting performance
1
Client Hello
- Supported ciphers
- Random number
- Session ID (if any)
Server Hello
2
- choosen cipher
- random number
- Session ID (Re-use, or new)
Check certificate validity
Server Certificates
3
Server Hello Done
4
5
ClientKey Exchange Message
- Encrypted pre-master secret
6
Key generation
Key generation
6
CipherSpec Exchange
7
Finished
CipherSpec Exchange
Finished
7
Encrypted data
ASYMMETRIC KEY SIZE
ASYMMETRIC ALGORITHM
SERVEUR HARDWARE
AES128
AES256
RC4
...
SYMMETRIC ALGORITHM