链滴

# HTTPS 详解

作者：zhengliwei

原文链接：https://ld246.com/article/1602382240361

来源网站：链滴

hello，大家好，欢迎来到银之庭。我是Z，一个普通的程序员。今天我们来聊聊HTTPS协议。

# 1.HTTPS简介

任何一种技术都是为了解决某些问题存在的，HTTPS也一样，它是为了解决HTTP协议的一些问题而创造出来的，我们先来看看HTTP存在的问题。

## 1.1 HTTP协议的问题

1. 内容 **明文传输**，容易在传输过程中被截获，直接拿到传输的敏感数据。
2. 数据发送和接受双方都没有做 **数据校验**，在传输过程中数据容易被他人篡改。
3. 浏览器无法 **确认服务器是不是正规服务器**，可能他人伪造的服务器。

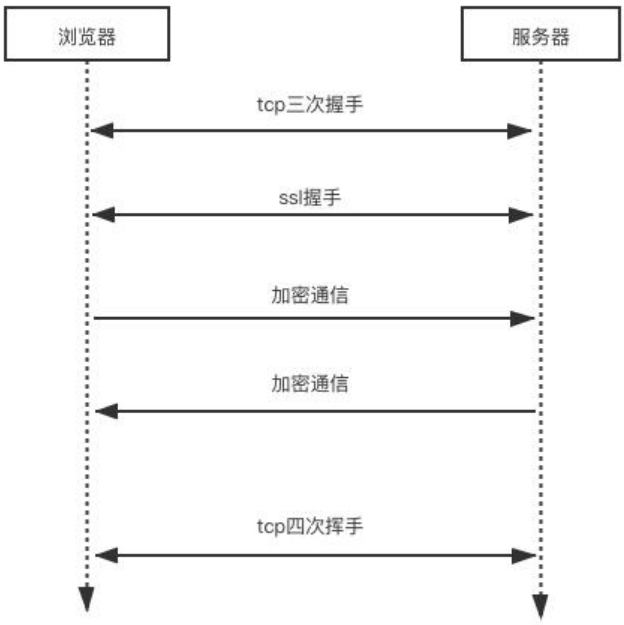
## 1.2 HTTPS的解决方案

为了解决HTTP协议存在的各种问题，HTTPS协议应运而生。它实际上是在HTTP协议之下，TCP协议上加了一层SSL或TLS协议（SSL和TLS协议很类似，以下内容以SSL协议为例），用于加密HTTP协议输的内容，另外，它还会在浏览器与服务器建立连接的过程中增加几步，来实现SSL加密通信的功能及让浏览器验证服务器身份。对应上述的HTTP的问题，HTTPS的解决方案如下：

1. 通过SSL协议 **加密**传输内容，避免在传输过程中被他人获取敏感数据。
2. 数据报文中 **增加内容校验和**，接收方验证校验和，避免在传输过程中数据被他人修改。
3. 在浏览器和服务器建立连接时，浏览器会先 **校验服务器的证书**，确保服务器是正规服务器。

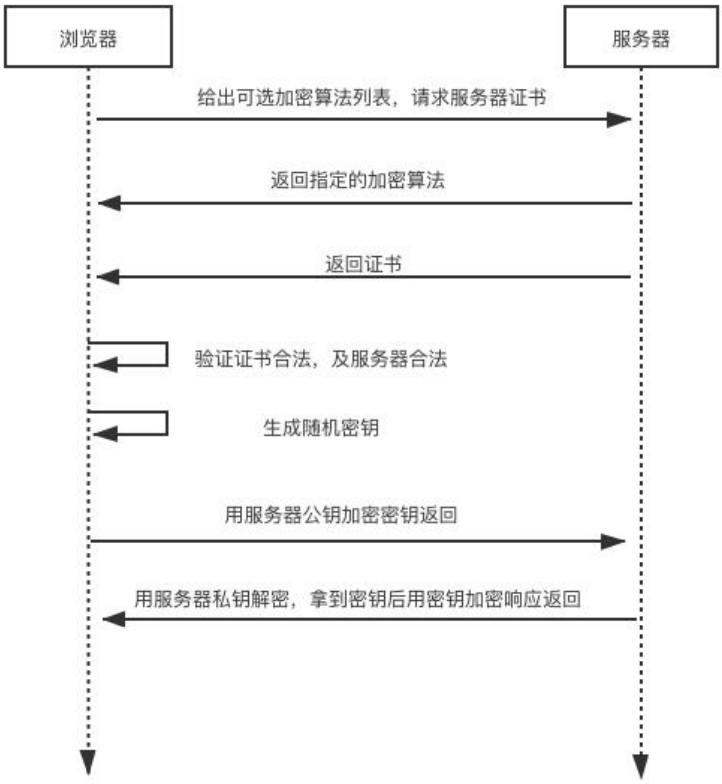
# 2. HTTPS通信过程

使用HTTPS进行的浏览器和服务器的通信过程简化版如下图：




图中我们简化了TCP三次握手和四次挥手的过程，那不是本文的重点。从上图可以看出，相比HTTP，

TTPS协议在建立连接时多了SSL握手的过程，并且在数据传输过程中一直以加密形式进行数据传输。面我们详细介绍下SSL握手的过程，如下图：



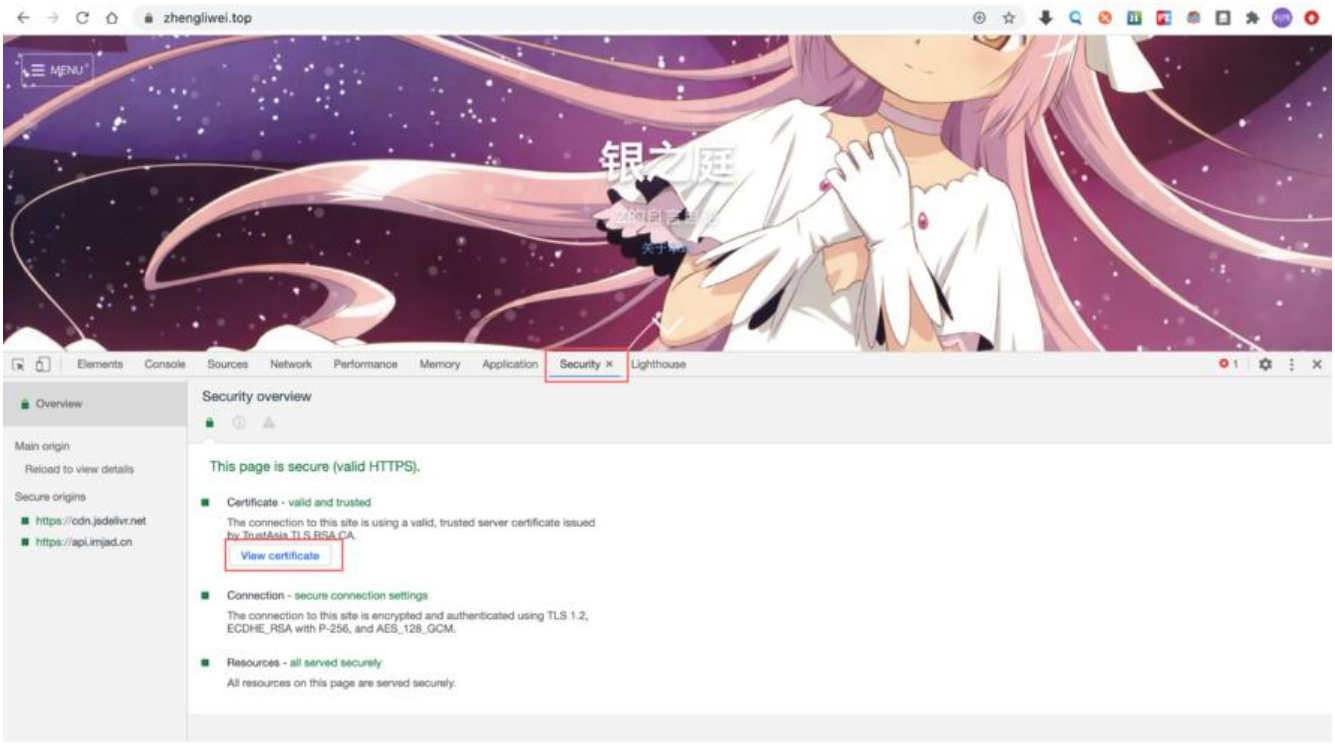

HTTPS使用了对称加密和非对称加密结合的方式进行通信，具体地说就是用非对称加密的方式约定一密钥，然后用这个密钥进行对称加密通信。这样做的目的是结合两者的优点，避免两者的缺点：对称密性能更高，但密钥的传递过程不安全，容易被窃取；非对称加密安全性更高，但性能较差。所以，非对称加密来传递密钥，保证了密钥的安全性后就可以用性能更高的对称加密来进行数据传输了。以是建立连接的详细交互过程：

1. 浏览器先向服务器的443端口发送请求，给出自己支持的加密算法列表，让服务器选择，即约定一后续加密使用的算法。

2. 服务器选定一个加密算法返回给浏览器。

3. 服务器向浏览器发送自己的证书和公钥。证书中包含很多信息，主要的有：证书颁发机构（ca机构，证书有效期，证书拥有者的信息，证书颁给的域名以及证书的签名等。想查看一个网站的证书都含哪些内容，可以在Chrome的开发者工具中切换到security页面，点击“view certificate“按钮，就看到该网站的证书详细内容了，以银之庭为例：

原文链接：HTTPS 详解




4. 浏览器验证证书和服务器的合法性。这一步很关键，详细步骤为：浏览器先从服务器证书中查询证颁发机构，先验证该机构的证书是否可信，而该机构的证书又会有个上级机构的证书，这样一直顺着书链查找，直到找到根证书，验证根证书的机构是否可信，可信的机构列表已经内置在浏览器里了，以浏览器可以直接验证，如果根证书不在信任名单里，浏览器会给出提示，如果在，则可以证明这条书链没有问题，可以信任服务器的证书。接着，浏览器会用证书的ca机构的公钥解密证书的签名，得证书的指纹和指纹算法，然后用指纹算法对证书内容进行指纹计算，如果和签名拿到的指纹一样，则以证明证书内容没有被修改过（假如中间人拿到了证书，进行了修改，由于没有ca机构的私钥，无法成新的签名，所以浏览器会发现新计算的指纹和签名中的指纹对不上，就会发现证书被修改了）。接验证一下证书其他字段是否合法，如是否在有效期内，颁发给的域名是否和正在请求的域名一致（避被跳转到钓鱼网站，钓鱼网站可能有证书，但证书里的域名肯定和原网站的域名不同，所以浏览器会验不通过）等。

5. 在一切校验通过后，浏览器会生成随机的密钥，然后用服务器的公钥加密，发送给服务器。

6. 服务器用自己的私钥解密内容，拿到密钥，至此密钥约定完成，后续就可以直接用对称加密进行通了。

# 3. HTTPS的问题

HTTPS协议并不是完美的，毕竟，计算机领域一个永恒不变的话题就是折中。

HTTPS相比HTTP可以增加了加密过程，不可避免地会导致服务器资源消耗增加，另外在建立连接时要约定加密密钥，这个步骤也会增加连接耗时。不过，相信浏览器也做了很多优化来减少这方面的影，具体实现我就不是很清楚了。

在安全性上，虽然在理论上HTTPS的安全性是足够高的，但由于浏览器是人在操作，如果人的操作不全，也会破坏HTTPS协议的安全性，如浏览器使用者手动信任了一个浏览器不信任的证书，就有可能中间人代理通信过程，实际上在APP开发中常用的Charles抓包的过程就是使用中间人代理客户端和务器的通信过程，在配置Charles抓包时会需要我们手动安装并信任Charles的证书，这一步就是很危的操作。

# 4. 推荐阅读

推荐几篇我个人认为写的不错的文章：

https://www.jianshu.com/p/f6b34381beac

https://blog.csdn.net/liuxingrong666/article/details/83869161

https://segmentfault.com/a/1190000022662058

以上，就是我目前对HTTPS协议的全部理解了，说实话，这篇文章的质量我并不是很满意，它只是停在原理层面进行了介绍，没有深入到实际操作中去，比如实际抓包看一下浏览器和服务器的ssl握手过等，以及对证书的生成和验证等方面介绍不够详细，但实在是时间有限，无法查阅更多资料，以后有会我会再补充这篇文章的。就这样 ~