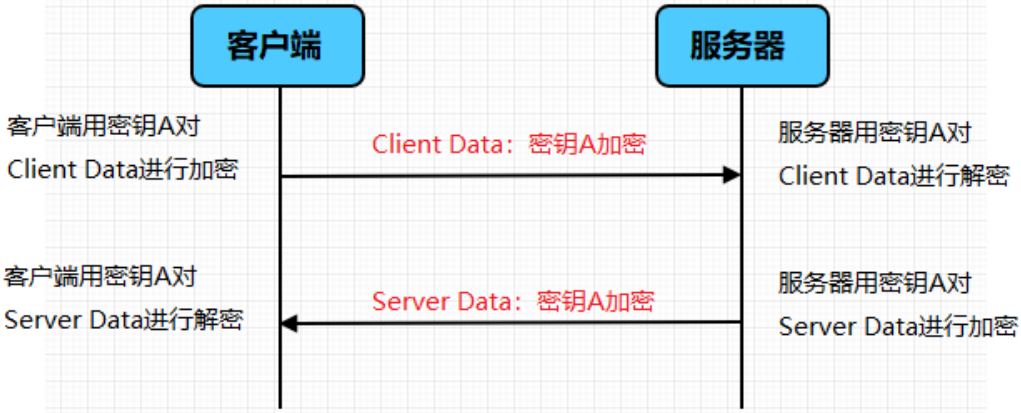
HTTPS详解

一、 HTTPS概述

HTTPS（超文本传输安全协议），是以安全为目标的HTTP通道，简单讲是HTTP的安全版 。 即HTTP下加入SSL层，HTTPS的安全基础是SSL 。WEB服务存在http和https两种通信方式，http默认采用80作为通讯端口，对于传输采用不加密的方式；https默认采用443，对于传输的数据进行加密传输。目前主流的网站基本上开始默认采用HTTPS作为通信方式。

二、 几个概念

1、 对称加密



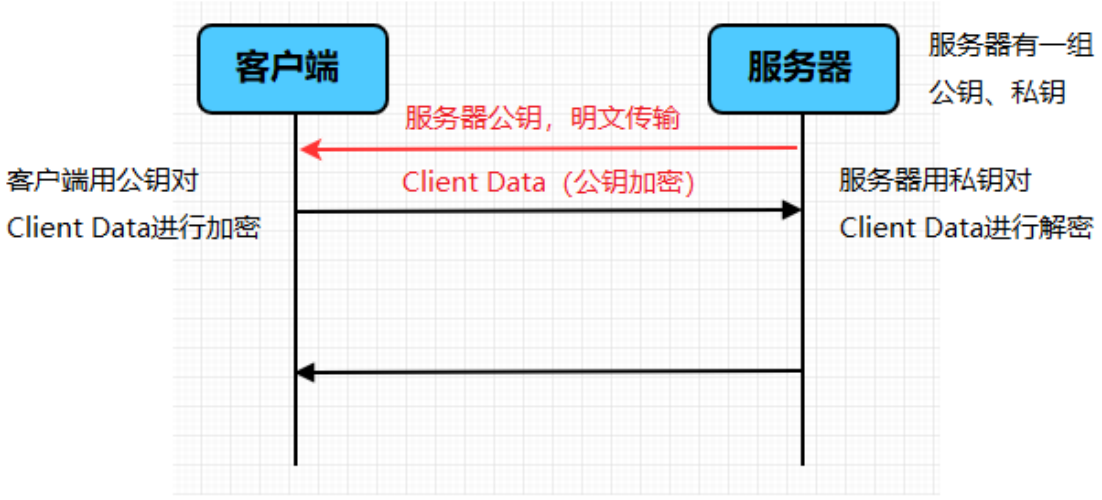
对称加密算法的加密和解密都是用**同一个密钥。**

如果通信双方都各自持有同一个密钥，且没有别人知道，则两方的通信安全是可以被保证的（除非密钥被破解）。

然而，最大的问题就是这个密钥怎么让传输的双方知晓，同时不被别人知道。如果由服务器生成一个密钥并传输给浏览器，这个传输过程中密钥被别人劫持，之后他就能用密钥解开双方传输的任何内容。

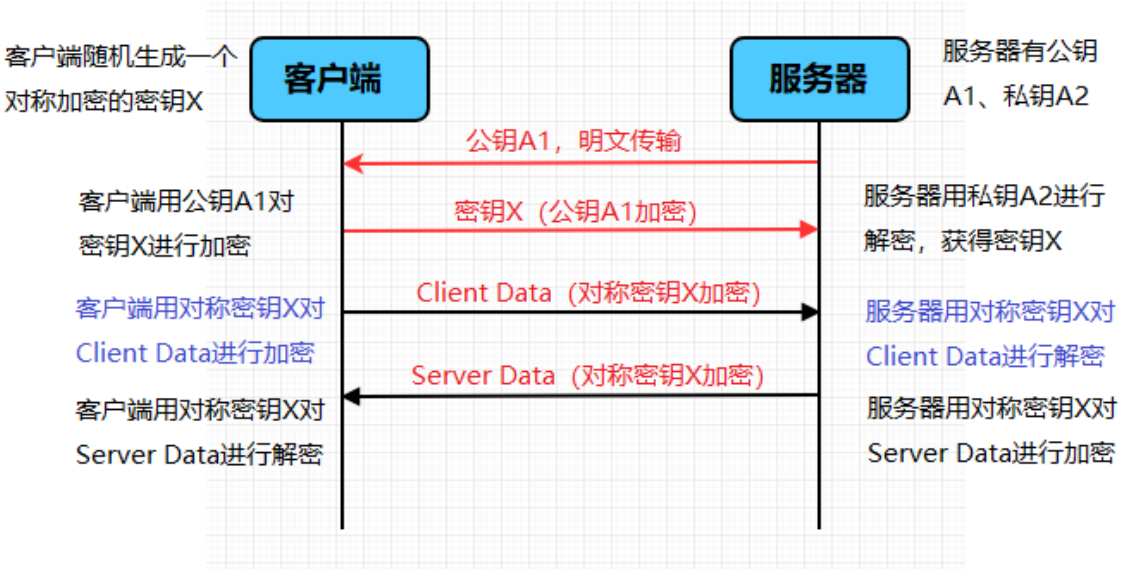
如果浏览器内部预存了网站A的密钥，且可以确保除了浏览器和网站A，不会有任何外人知道该密钥，那理论上用对称加密是可以的。这样，浏览器只要预存好世界上所有HTTPS网站的密钥就可以了。显然，这样做是不现实的。所以解决这个问题，我们就需要非对称加密

2、 非对称加密



基于对称加密存在的问题，又有了非对称加密。非对称加密算法需要**一组密钥对**，分别是**公钥**和**私钥**，这两个密钥是成对出现的。**公钥加密的内容需要对应的私钥解密，私钥加密的内容需要对应的公钥解密。**私钥由服务器自己保存，**公钥发送给客户端。**客户端拿到公钥后可以对请求进行加密后发送给服务端，这时候就算中间被截获，没有私钥也无法解密发送的内容，这样确保了**客户端发送到服务端数据的安全。**

3、 非对称加密+对称加密



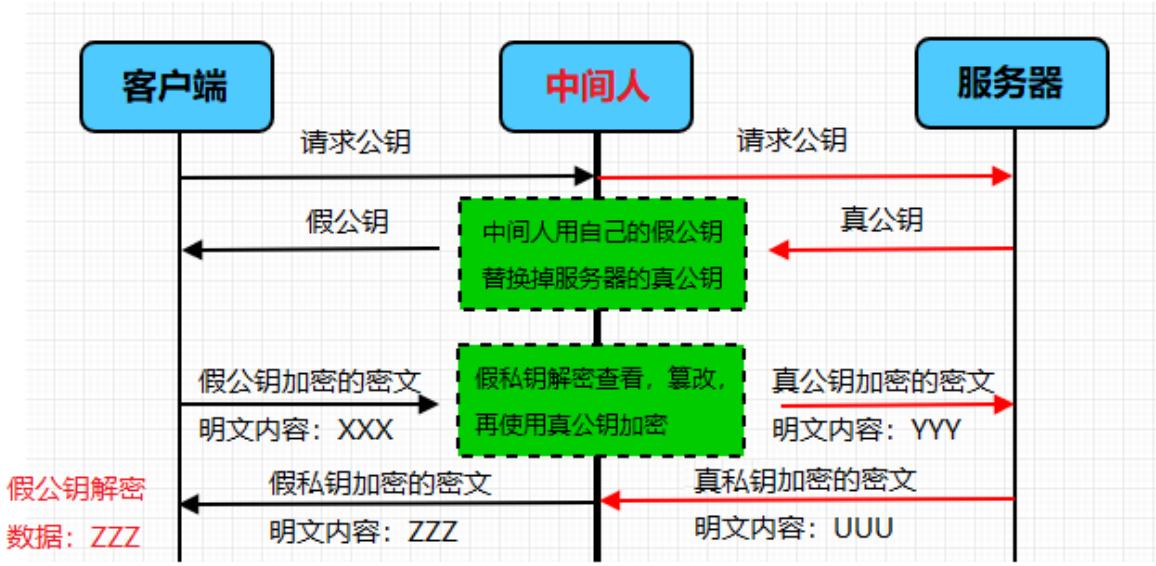
由于非对称加密算法非常耗时，特别是加密解密一些较大数据的时候有些力不从心。而对称加密快很多，我们考虑是否可以采用非对称加密+对称加密结合的方式，而且要尽量减少非对称加密的次数。

非对称加密、解密各只需一次的方法：

1. 某网站拥有用于非对称加密的公钥A1、私钥A2。
2. 浏览器向网站服务器请求，服务器把公钥A1明文给传输浏览器。
3. 浏览器随机生成一个用于对称加密的密钥X，用公钥A1加密后传给服务器。
4. 服务器拿到后用私钥A2解密得到密钥X。
5. 这样双方就都拥有密钥X了，且别人无法知道它。之后双方所有数据都用密钥X加密解密即可。

HTTPS基本就是采用了这种方案。但还是有漏洞的。

4、 中间人攻击



中间人的确无法得到浏览器生成的对称密钥X，这个密钥本身被公钥A1加密，只有服务器才能用私钥A2进行解密。然而中间人却完全不需要拿到私钥A2就能劫持信息，请看：

1. 某网站拥有用于非对称加密的公钥A1、私钥A2。
2. 浏览器向网站服务器请求，服务器把公钥A1明文传输给浏览器。
3. 中间人劫持到公钥A1，保存下来，把数据包中的公钥A1替换成自己伪造的公钥B1（它当然也拥有公钥B1对应的私钥B2）。
4. 浏览器随机生成一个用于对称加密的密钥X，用公钥B1（浏览器不知道公钥被替换了）加密后传给服务器。
5. 中间人劫持后用私钥B2解密得到密钥X，再用公钥A1加密后传给服务器。
6. 服务器拿到后用私钥A2解密得到密钥X。

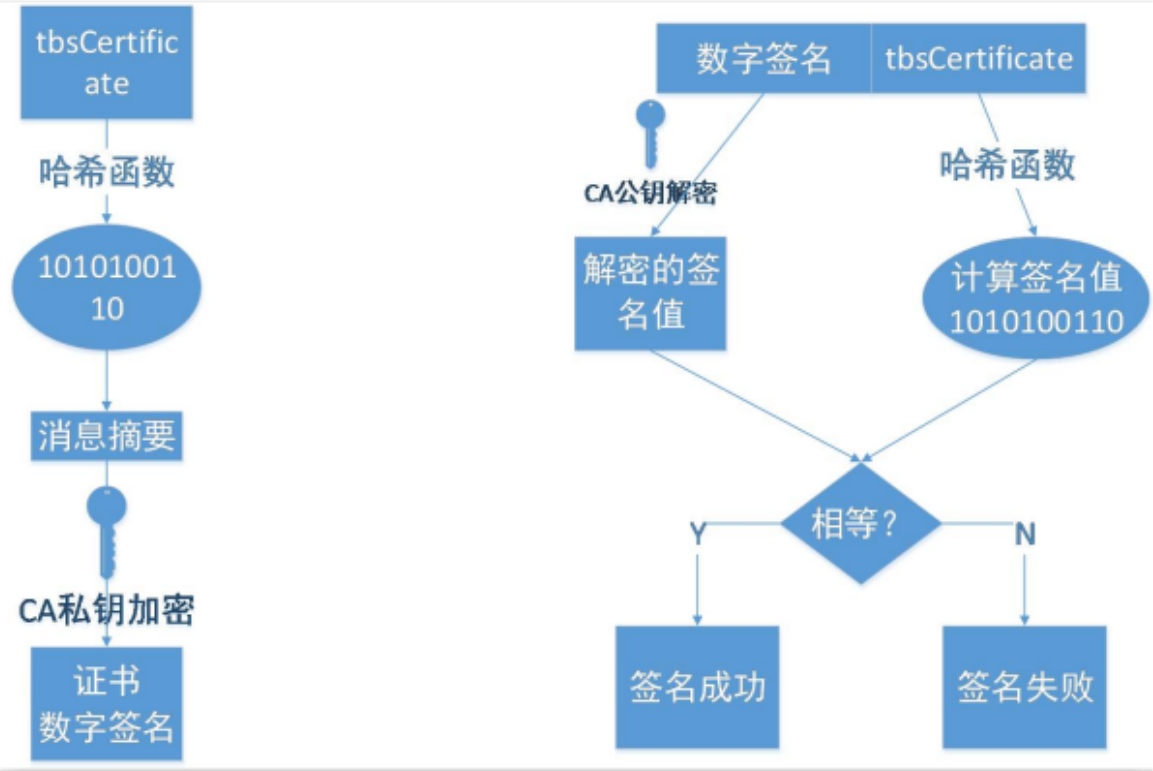
这样在双方都不会发现异常的情况下，中间人得到了对称密钥X。根本原因是浏览器无法确认自己收到的公钥是不是网站自己的。那么下一步就是解决这个问题：如何证明浏览器收到的公钥一定是该网站的公钥？

5、 数字证书

网站在使用HTTPS前，需要向“**CA机构**”申请颁发一**数字证书**，数字证书里有**证书持有者、证书持有者的公钥等信息**。服务器把证书传输给浏览器，浏览器从证书里取公钥就可以了。然而这里又有一个显而易见的问题：证书本身的传输过程中，如何防止被篡改？即如何证明证书本身的真实性？数字证书怎么防伪呢？

6、 数字签名

我们把证书内容生成一份“签名”，比对证书内容和签名是否一致就能察觉是否被篡改。这种技术就叫数字签名。



（左侧是数字签名的制作过程，右侧是验证过程）

数字签名的制作过程：

1. CA拥有非对称加密的私钥和公钥。
2. CA对证书明文信息进行hash。
3. 对hash后的值用私钥加密，得到数字签名

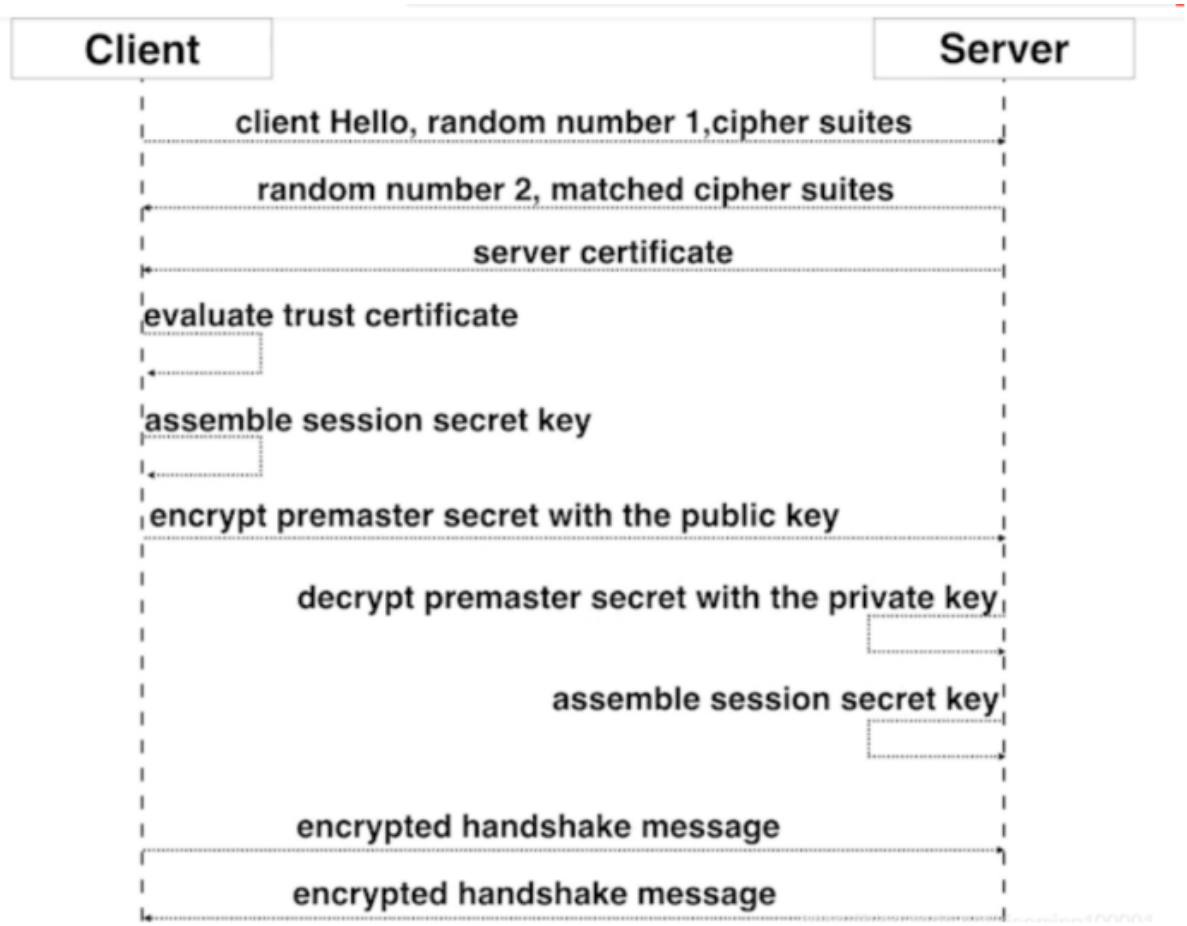
浏览器验证过程：

1. 拿到证书，得到明文T1，数字签名S1。
2. 用CA机构的公钥对S1解密（由于是浏览器信任的机构，所以浏览器保有它的公钥），得到hash值S2。
3. 用证书里说明的hash算法对明文T1进行hash得到T2。
4. 比较S2是否等于T2，等于则表明证书可信。

为什么这样可以证明证书可信?

假设中间人篡改了证书的原文，由于他没有CA机构的私钥，所以无法得到此时加密后签名，无法相应地篡改签名。浏览器收到该证书后会发现原文和签名解密后的值不一致，则说明证书已被篡改，证书不可信，从而终止向服务器传输信息，防止信息泄露给中间人。

三、 HTTPS 工作原理



1. client向server发送请求https://baidu.com，然后连接到server的443端口，发送的信息主要是随机值1和客户端支持的加密算法。
2. server接收到信息之后给予client响应握手信息，包括随机值2和匹配好的协商加密算法，这个加密算法一定是client发送给server加密算法的子集。
3. 随即server给client发送第二个响应报文是数字证书。服务端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，这套证书其实就是一对公钥和私钥。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、服务端的公钥，第三方证书认证机构(CA)的签名，服务端的域名信息等内容。
4. 客户端解析证书，验证是否有效，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，那么就生成一个随机值（预主秘钥）。
5. 客户端认证证书通过之后，接下来是通过随机值1、随机值2和预主秘钥组装会话秘钥。然后通过证书的公钥加密预主秘钥并发送给服务器。
6. 服务端解密得到预主秘钥，然后将随机值1，随机值2和预主密钥组装成会话秘钥，跟客户端会话秘钥相同。
7. 客户端通过会话秘钥加密一条消息发送给服务端，主要验证服务端是否正常接受客户端加密的消息。
8. 同样服务端也会通过会话秘钥加密一条消息回传给客户端，如果客户端能够正常接受的话表明连接建立完成了。

四、 https和http的区别

SSL证书需要购买申请，功能越强大的证书费用越高

SSL证书通常需要绑定IP，不能在同一IP上绑定多个域名，IPv4资源不可能支撑这个消耗

根据ACM 的实验数据显示，使用HTTPS协议会使页面的加载时间延长近50%，增加10%到20%的耗电。

HTTPS连接缓存不如HTTP高效，流量成本高。

HTTPS连接服务器端资源占用高很多，支持访客多的网站需要投入更大的成本。

HTTPS协议握手阶段比较费时，对网站的响应速度有影响，影响用户体验。比较好的方式是采用分而治之，类似12306网站的主页使用HTTP协议，有关于用户信息等方面使用HTTPS