白话图解 HTTPS 原理

【前言】最近看过几篇文章，内容是关于“全民 HTTPS”的。为什么 HTTPS，突然会受到如此多业内人士的青睐呢？HTTPS 究竟是什么呢？它与 HTTP 又有怎样的区别呢？

带着这个问题，我查看了很多网上的资料，但是太多的专业词汇，让我感到头疼，很难理解。按我个人的认知，这个世界上根本不存在任何高深的道理，尤其是西方讲究科学性的思维，再高深，也不可能高深过中国传统思想中的八个字“只可意会，不可言传”，完全不该诉你，让你自己去领悟，这多难啊。

所以，我又耐着性子，参照翟志军的《也许，这样理解 HTTPS 更容易》一文，以及百度词条的相关定义，用类比的方法，追根溯源，重新梳理一下，才恍然大悟。本着分享的精神，我就把自己所感所悟的过程记录下来，希望大家读罢此文，也能有所感悟！

【思考一：为什么 HTTPS 会突然受到青睐？】

答案就四个字——HTTP 劫持！

【思考二：什么是 HTTP 劫持？】

官方解释：什么是 HTTP 劫持呢，大多数情况是运营商 HTTP 劫持，当我们使用 HTTP 请求请求一个网站页面的时候，网络运营商会在正常的数据流中插入精心设计的网络数据报文，让客户端（通常是浏览器）展示“错误”的数据，通常是一些弹窗，宣传性广告或者直接显示某网站的内容。

通俗解释：你预定了某类军事杂志（类比“客户端向服务器发送了一条请求，访问军事网站”），此时黑心快递员（类比“不要脸的运行商或黑客”）拆封了你的快递（类比“劫持了你的正常数据流”），然后在军事杂志的每一页，都贴上了“包小姐”的小广告（类比“精心设计的网络数据报文”），再送行打包送给你，获得额外盈利。

【引申：什么是 DNS 劫持？】

官方解释：DNS 劫持就是通过劫持了 DNS 服务器，通过某些手段取得某域名的解析记录控制权，进而修改此域名的解析结果，导致对该域名的访问由原 IP 地址转入到修改后的指定 IP，其结果就是对特定的网址不能访问或访问的是假网址，从而实现窃取资料或者破坏原有正常服务或诈骗钱财的目的。

通俗解释：高考填报考时，你在志愿书上填写了“北大”（类比“正常请求”），但是你的前任偷走了你的志愿书（类比“劫持了 DNS 服务器”），把他改成了“北大某鸟”。最后的结果就是，你以 690 分的成绩考入了北大某鸟（类比“最终访问了假网址”），然后还伪装北大招生办，让你往某银行卡打入 5 千元学费（类比“后续诈骗钱财”）。

【一句话概括：HTTP 劫持与 DNS 劫持的区别？】

HTTP 劫持：你打开的是人民日报的官方网站，右下角却弹出了蓝翔的挖掘机广告。

DNS 劫持：你在地址栏输入的是人民日报的网页地址，却打开了淘宝特卖的钓鱼网站。

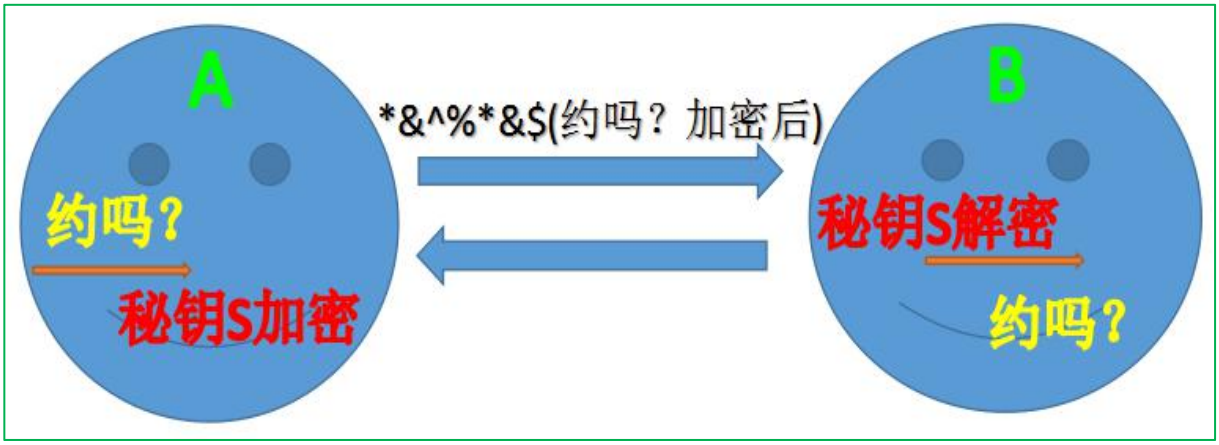
【思考三：HTTP 为什么会被劫持？】

一句话：HTTP 在传输的过程中使用的是明文！

通俗解释：你在快递东西的时候，快递员是可以看到你的物品的，这样，他就可以随意更换篡改你的东西。

【思考四：如何避免 HTTP 明文的不良影响呢？（推演 HTTPS 协议的前世今生）】

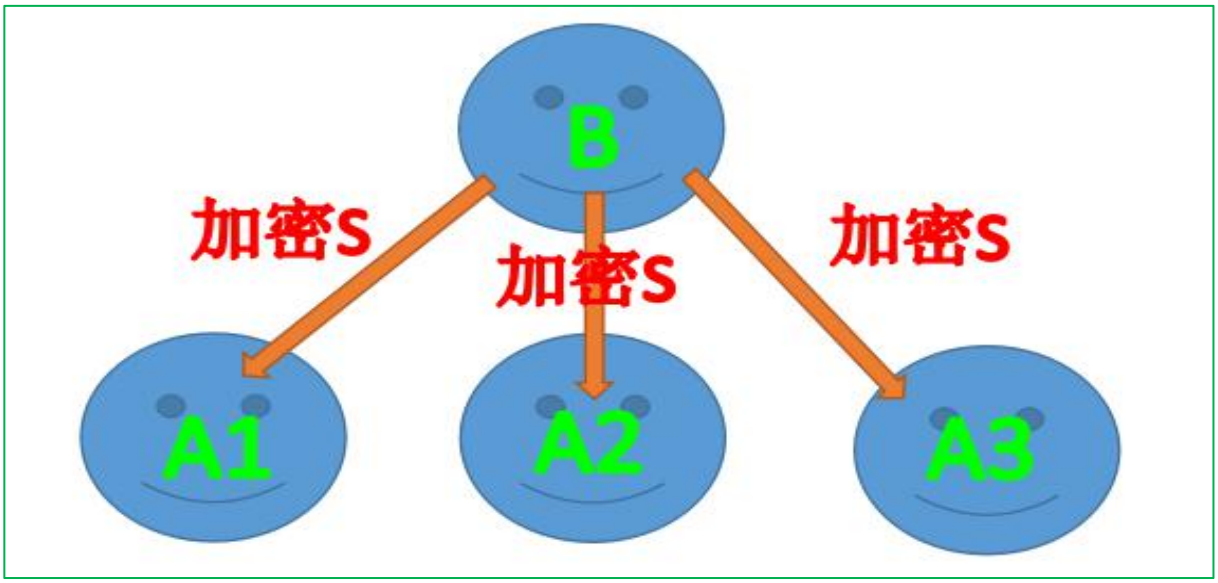
1.最先想到的就是对明文进行加密，比如各种对称加密（DES）；



安全成立的条件：S 足够安全，且只有 A 和 B 知道；

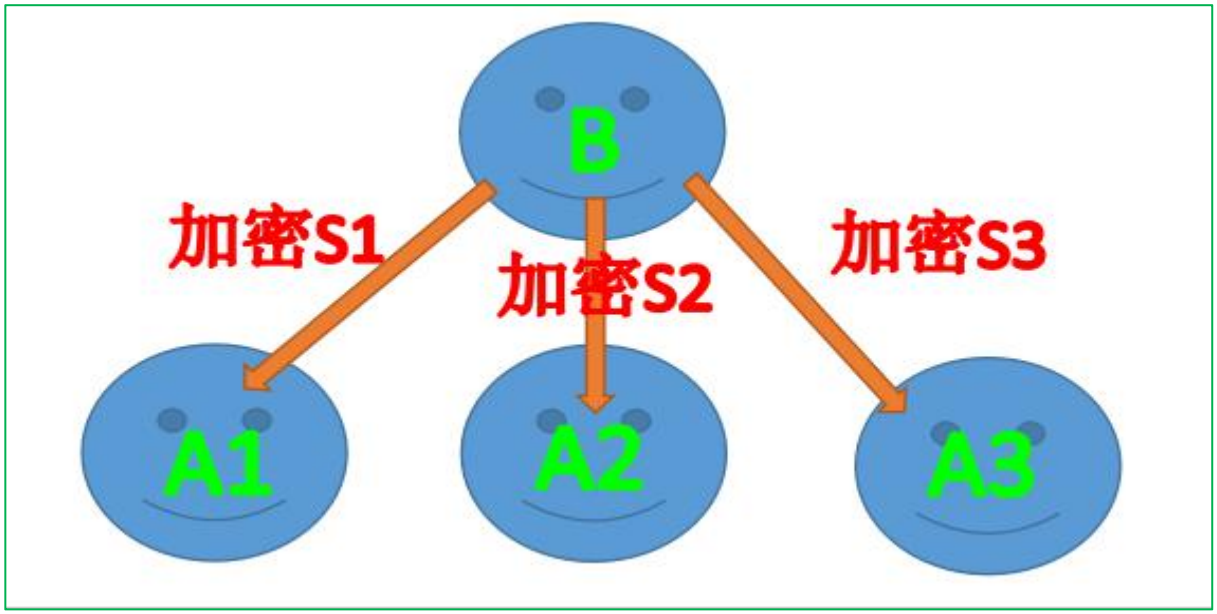
但是这样就可以了吗？——远没想象那么简单！

如果服务器端 B 对所有的客户端通信 A 都使用同样的对称加密算法 S，那么，这是不是就可以轻易破解，也就无异于没有加密了呢？



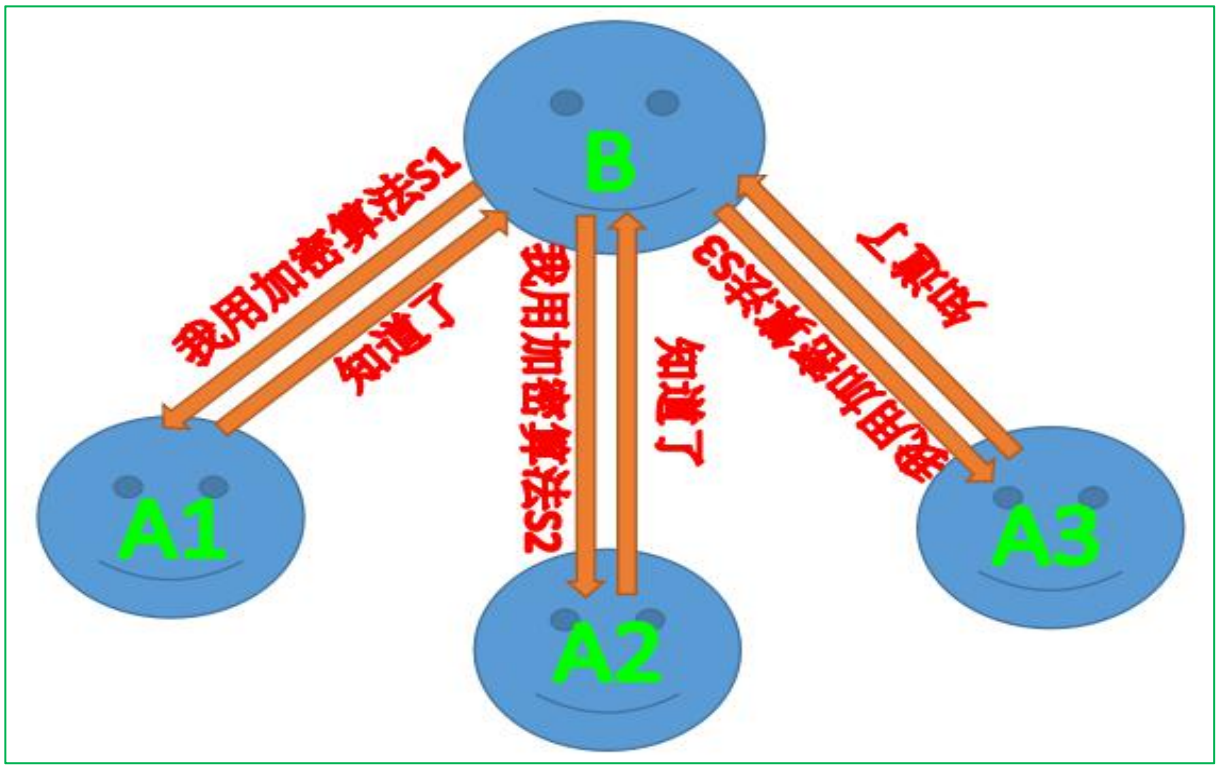
2.如何即能使用对称加密算法，又不公开密钥？

答：Web 服务器与每个客户端使用不同的对称加密算法：



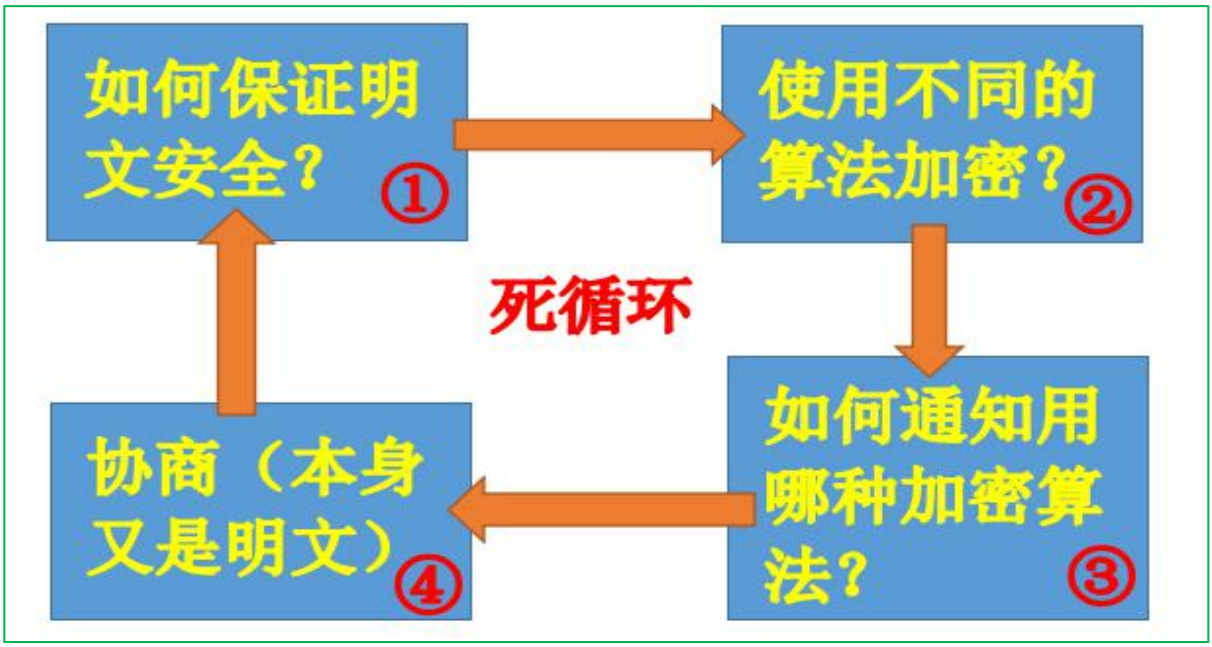
3.服务器端 B 怎么告诉客户端 A 该使用哪种对称加密算法 S 呢？

答：通过协商。



但是，这样就安全了吗？——依然没有，因为这个协商过程，本身又是裸露的，依然可以被获知，那么怎么办呢？

有人说继续对“协商内容”进行加密，那么，此时你有没有发现，已经陷入一个死循环了呢？



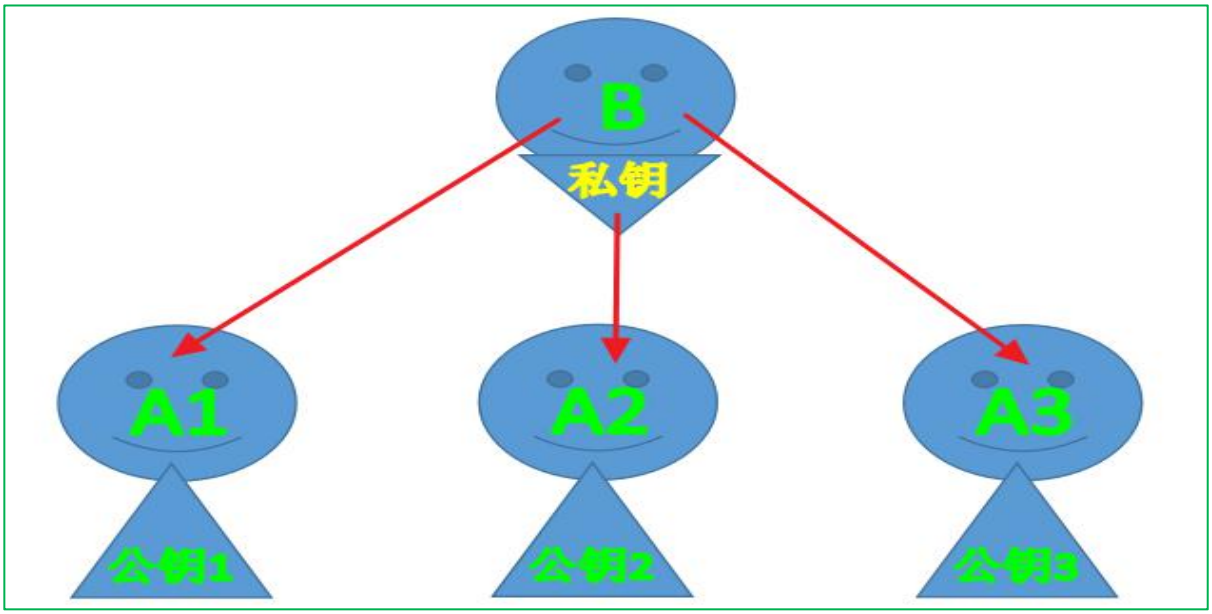
4.如何解决这种对称加密的死循环呢？

答：用非对称加密（常用 RSA）。

5.什么是非对称加密？

官方解释：对称加密算法在加密和解密时使用的是同一个秘钥；而非对称加密算法需要两个密钥来进行加密和解密，这两个秘钥是公开密钥（public key，简称公钥）和私有密钥（private key，简称私钥）。

通俗解释：政府的财政收入（类比“私钥加密后的密文”），只要是公民，都有权知道(类比“只要是公钥，都可以解密”)；但是公民个人的收入（类比“公钥加密后的密文”），除了自己，只有政府才可以知道，其他人不行（类比“只有私钥可以解密”）。中国政府只有一个，但公民是有很多的(类比“私钥只有一个人有，而公钥可以发给所有的人”)。



此时你应该明白了：HTTPS 同时需要对称加密算法和非对称加密算法。

6.要达到 Web 服务器针对每个客户端使用不同的对称加密算法，同时，我们也不能让第三者知道这个对称加密算法是什么，怎么办？

答：使用随机数。就是使用随机数来生成对称加密算法。这样就可以做到服务器和客户端每次交互都是新的加密算法、只有在交互的那一该才确定加密算法。

此时，你明白为什么 HTTPS 协议握手阶段会有这么多的随机数了吧。

7.如何让客户端安全地得到公钥呢？

答：两种假设：

①让服务器端将公钥放到一个远程服务器，客户端可以请求得到；

②让服务器端将公钥发送给每一个客户端；

先看假设①，将公钥放到一个远程服务器，客户端又需要发送请求才能得到。注意，一旦有请求，又有明文，又回到那个死循环了。显然不可以！

再看假设②,仍然有一个问题：如果服务器端发送公钥给客户端时，被中间人调包了，怎么办？显然不可以！

为了方便能理解，请看下图：

8.如何解决公钥被调包的隐患呢？

公钥被调包的问题出现，是因为我们的客户端无法分辨返回公钥的人到底是中间人，还是真的服务器。这其实就是密码学中提的身份验证问题。

问题的关键在于：我们选择直接将公钥传递给客户端的方案，我们始终无法解决公钥传递被中间人调包的问题。

所以，我们不能直接将服务器的公钥传递给客户端，而是第三方机构使用它的私钥对我们的公钥进行加密后，再传给客户端。客户端再使用第三方机构的公钥进行解密，这样就引出了数字证书的概念。

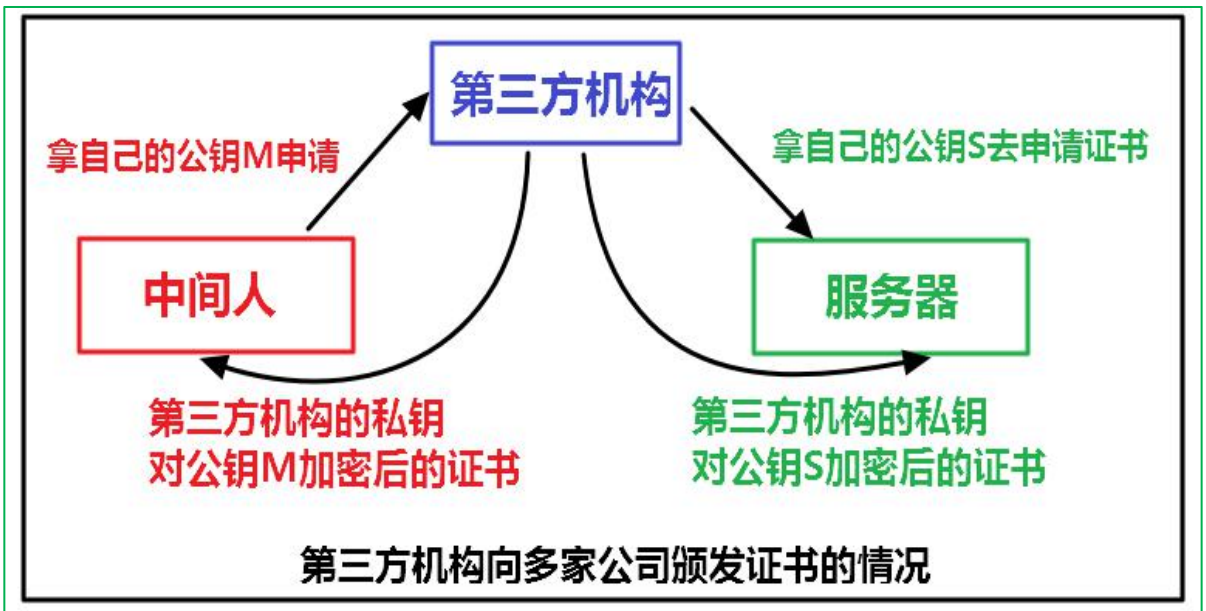
9.什么是数字证书？

官方解释：数字证书是一个经证书授权中心数字签名的包含公开密钥拥有者信息以及公开密钥的文件。最简单的证书包含一个公开密钥、名称以及证书授权中心的数字签名。数字证书还有一个重要的特征就是只在特定的时间段内有效。数字证书是一种权威性的电子文档，可以由权威公正的第三方机构，即 CA（例如中国各地方的 CA 公司）中心签发的证书，也可以由企业级 CA 系统进行签发。

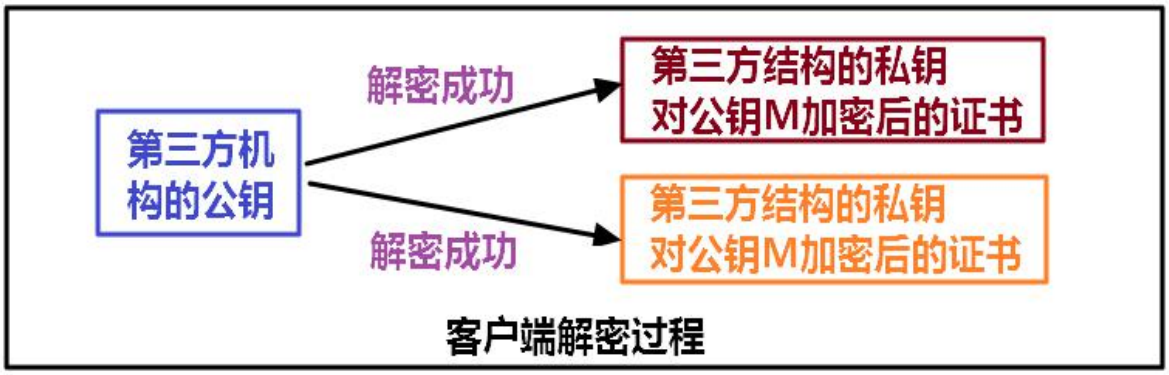
说明：证书中只有服务器交给第三方机构的公钥，而且这个公钥被第三方机构的私钥加密了。如果能解密，就说明这个公钥没有被中间人调包。因为如果中间人使用自己的私钥加密后的东西传给客户端，客户端是无法使用第三方的公钥进行解密的。

但是第三方机构不可能只给你一家公司制作证书，它也可能会给中间人这样有坏心思的公司发放证书。这样的，中间人就有机会对你的证书进行调包，客户端在这种情况下是无法分辨出是接收的是你的证书，还是中间人的。因为不论中间人，还是你的证书，都能使用第三方机构的公钥进行解密。

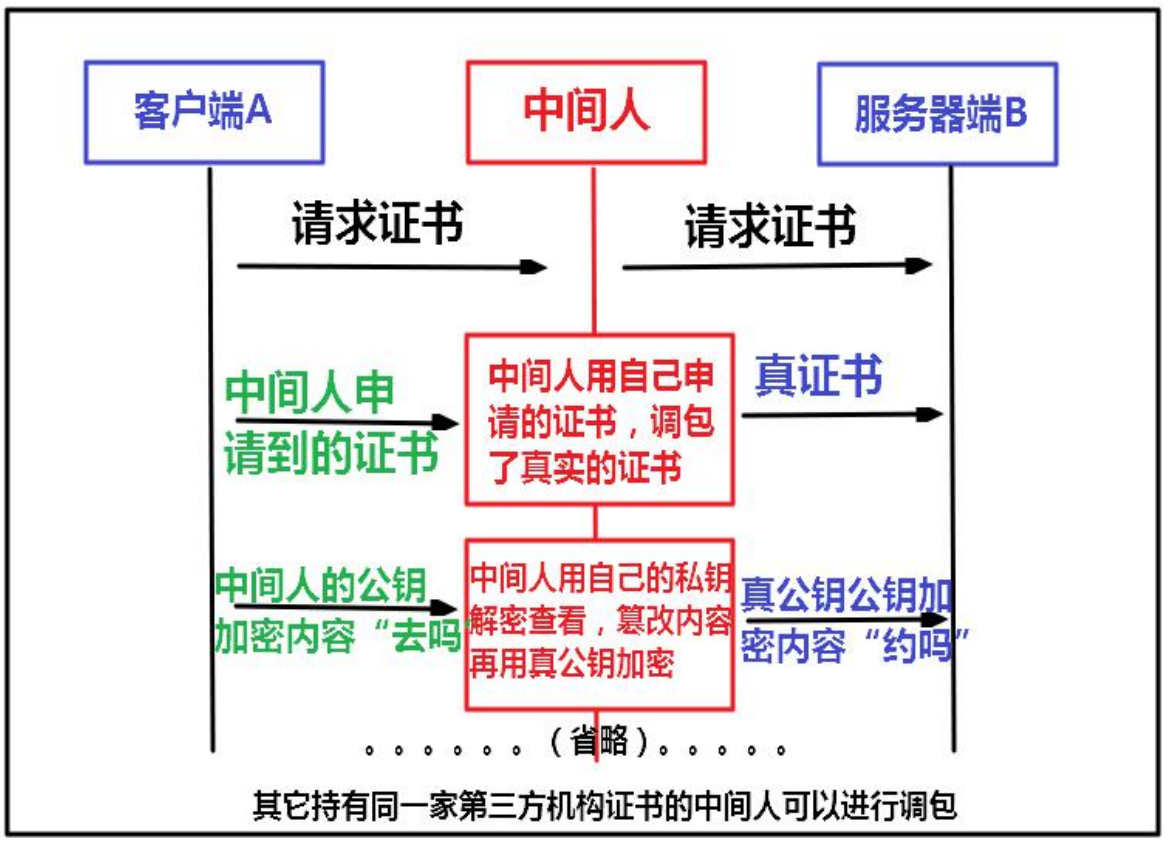
这样，又出现了不安全因素，像下面这样：



第三方机构向多家公司颁发证书的情况



客户端解密过程



其它持有同一家第三方机构证书的中间人可以进行调包

10. 如何解决同一机构颁发的不同证书被篡改的问题呢？

要解决这个问题，我们首先要想清楚 11 和 12 这两个问题：

11. 谁来负责辨别同一机构下不同证书呢？

答：客户端。客户端在拿到证书后，自己就有能力分辨证书是否被篡改了。

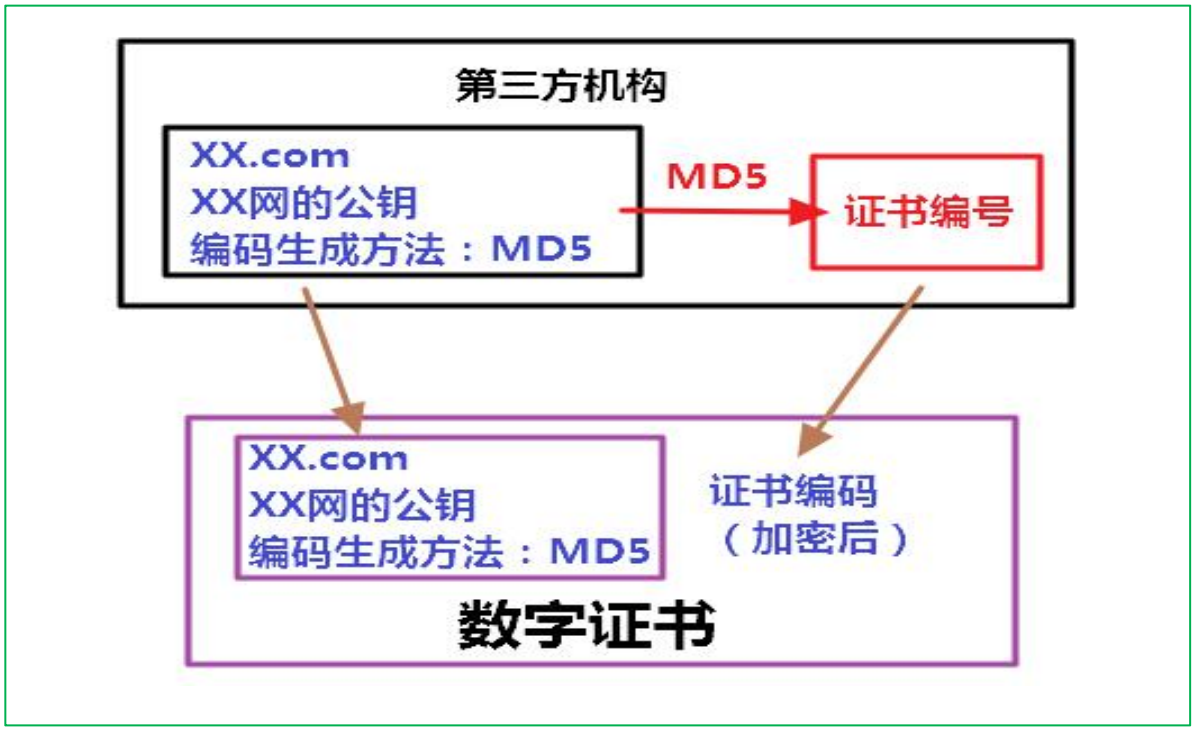
12. 如何才能拥有分辨证书是否被篡改的能力呢？

答：使用数字证书。我们从现实中找灵感。比如你是 HR，你手上拿到候选人的学历证书，证书上写了持证人，颁发机构，颁发时间等等，同时证书上，还写有一个最重要的：证书编号！我们怎么鉴别这张证书是的真伪呢？只要拿着这个证书编号上相关机构去查，如果证书上的持证人与现实的这个候选人一致，同时证书编号也能对应上，那么就说明这个证书是真实的。

13. 客户端本地怎么验证证书呢？

让证书本身写上“如何根据证书的内容生成证书编号”。也就是：客户端拿到证书后，根据证书上的方法自己生成一个证书编号，如果自己生成的证书编号与证书上的证书编号相同，那么说明这个证书是真实的。同时，为避免证书编号本身又被调包，所以使用第三方的私钥进行加密。

这地方有些抽象，我们来个图帮助理解：



当客户端拿到证书后，开始对证书中的内容进行验证，如果客户端计算出来的证书编号与证书中的证书编号相同，则验证通过。

14.这么多机器，第三方机构的公钥怎么跑到了客户端的机器中呢？

其实呢，现实中，浏览器和操作系统都会维护一个权威的第三方机构列表（包括它们的公钥）。因为客户端接收到的证书中会写有颁发机构，客户端就根据这个颁发机构的值在本地找相应的公钥。

【类比总结】

上边例子中：

所提到的**证书**,其实就是 **HTTPS 中的数字证书**；

所提到的**证书编号**就是 **HTTPS 中的数字签名**；

所提到的**第三方机构**就是指**数字证书签发机构（CA）**。

【一句话总结 HTTPS 的原理】

HTTPS 要使客户端与服务器端的通信过程得到**安全保证**，必须使用的**对称加密算法**，但是协商对称加密算法的过程，需要使用**非对称加密算**法来保证安全，然而直接使用非对称加密的过程本身也不安全，会有中间人篡改公钥的可能性，所以客户端与服务器不直接使用公钥，而是使用**数字证书签发机构（CA）**颁发的**证书**来保证非对称加密过程本身的安全，为了保证证书不被篡改，引入**数字签名**，客户端使用相同的对称加密算法,来验证证书的真实性，如此，最终解决了客户端与服务器端之间的通信安全问题。