





简单总结：

服务器端有非对称加密的私钥和公钥，通过明文发送公钥给浏览器，浏览器使用公钥进行数据加密，然后发送至服务器端，服务器利用对应的私钥进行解密。

问题1：服务器发送的公钥被篡改，怎么办？

浏览器 B 发送公钥请求到 服务端S，S返回公钥st给B，此时中间人M劫持到 st,使用自己的一对非对称加密的公钥mt和私钥ms，将私钥mt返回给B，B误以为mt是S提供的公钥，使用公钥mt加密发送数据（包括B自己的公钥bt）给S，被M获取，M利用自己的私钥ms解密B的数据，此时获取到B提交的数据和bt，然后M再用st加密数据和自己的公钥mt返回给服务器S，S将响应信息利用mt加密返回给B，被M劫持,M利用私钥ms解密响应信息，得到S响应数据，那么从而得到B,S的请求和响应信息，M可以改变任何通信中的数据而达到破坏原始数据。

根本原因是浏览器无法确认获取到的公钥到底是不是服务器的。CA机构做的事情就是给服务器的公钥提供类似人类身份证的数字证书。CA证书生成大概流程是：将服务器公钥利用hash算法得到hash值，然后利用CA自己的私钥对hash值进行加密，得到数字签名，从而服务器的明文公钥加上数字签名共同组成了数字证书。服务器将CA颁发的数字证书送给浏览器，浏览器通过验证服务器的证书来判断公钥是否被篡改。

问题2：浏览器如何验证？

浏览器得到数字证书，得到明文公钥T和数字签名S，用CA机构的公钥对S解密（由于CA是浏览器信任的机构，浏览器保有它的公钥），可以得到解密后的hash值，浏览器利用证书的hash算法对明文公钥进行hash得到一个hash值，对比2个hash值即可确认公钥是不是服务端的公钥。

问题3：怎么证明CA机构的公钥是可信的？

其实CA机构的公钥也是利用数字证书来证明的，原理一样的。操作系统或者浏览器会提前装一些信任的根证书，如果其中有该CA机构的根证书，那么就可以拿到它应的可信的公钥了。实际证书之间的认证也可以不止一层，可以A信任B，B信任C，以此类推，构成了一个信任链，也可称作数字证书链，以根证书为起点，通过层层信任，使终端实体证书的持有者可以获取转授的信任，以证明身份。

大概流程示意图如下：

