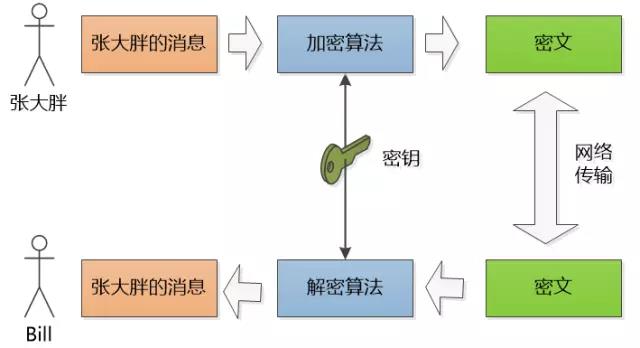
## 起因

在网络世界中，黑客可能监听任何一台电脑的通信信息。如果通信是明文的，未加密的，那么就有可能抓取我们的数据包，直接窥探个人隐私。为了保护网络世界的个人隐私，就需要对数据进行加密。那么怎么进行加密比较安全呢。现在我们就假设张大胖和bill要进行通信，为了防止黑客抓包窃取私人信息，我们就需要进行加密。

## 对称加密（加密和解密用同一个密钥）

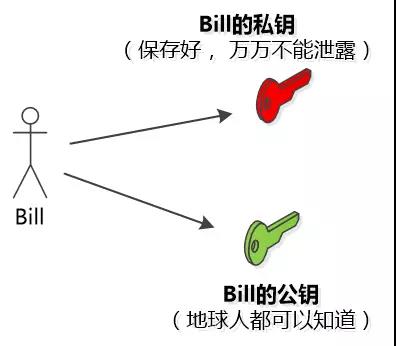
张大胖将传输数据通过密钥和加密算法进行加密，密钥只有通信的双方才知道，bill接收到数据通过密钥和解密算法进行解密。



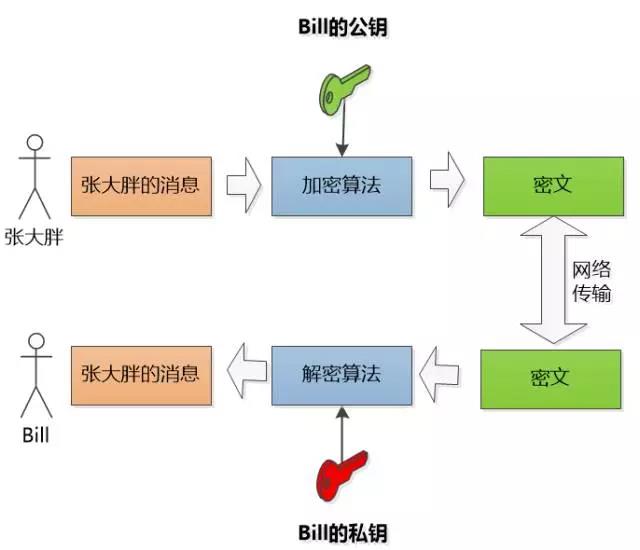
但是张大胖和bill通过网络发送密钥也会被黑客截取，黑客也能用密钥对密文进行解密。

## 非对称加密（RSA）

张大胖和bill每个人有两把钥匙，一把公钥，可以公开，一把私钥，只有自己知道。用私钥加密的数据，只有对应的公钥才能解密，用公钥加密的数据， 只有对应的私钥才能解密。



当张大胖给bill发信息时可以用bill的公钥进行加密，bill接收到数据用私钥解密。



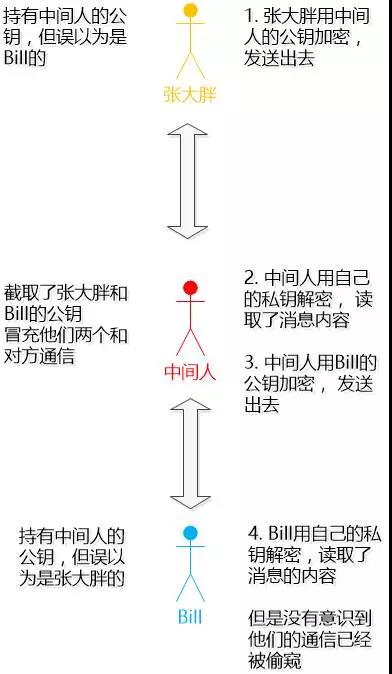
但是这个加密和解密非常耗费。

## 非对称加密+对称加密

为了加快加密解密过程，可以结合非对称加密和对称加密，通过非对称加密的方式传输对称加密的密钥，在通过对称加密进行通信明文的加密解密工作。

## 中间人攻击

但是紧接着问题又来了，当bill给张大牛发送公钥时，黑客可能会截取bill的公钥，并且给张大牛发送黑客自己的公钥，张大牛以为是bill的公钥，就用黑客的公钥进行加密，然后发送出去，黑客又截取到张大牛的密文用公钥进行解密得到明文内容，然后改动明文内容用bill的公钥进行加密发送给bill。



## 数字证书

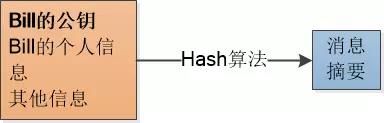
怎么确认公钥的拥有者？

现实中有公证处，它提供的公证材料大家都信任，那在网络世界也可以建立一个这样的具备公信力的认证中心， 这个中心给大家颁发一个证书， 用于证明一个人的身份。

这个证书包含有个人信息和公钥

但是怎么确保证书被安全传输而不被篡改呢？

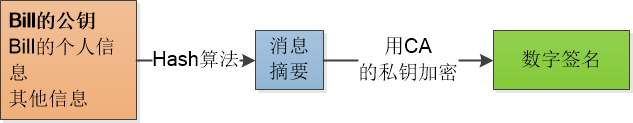
1. **通过hash算法为证书生成消息摘要，只要证书有一点改变生成的消息摘要就会有很大的改变。**



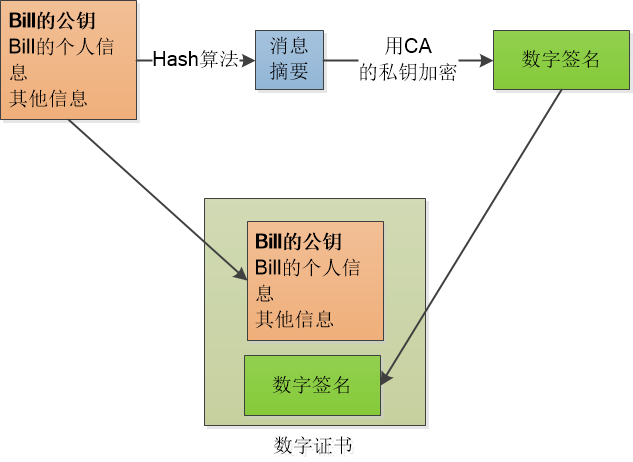
这样张大牛得到证书，用同样的hash算法为证书生成消息摘要，再对比发送过来的消息摘要查看是否一致就能知道是否被修改。但是攻击者虽然无法改变公钥，但是可以把整个原始信息都替换了， 生成一个新的消息摘要， 使张大牛无法辨别

这个时候就需要认证中心（CA）了。

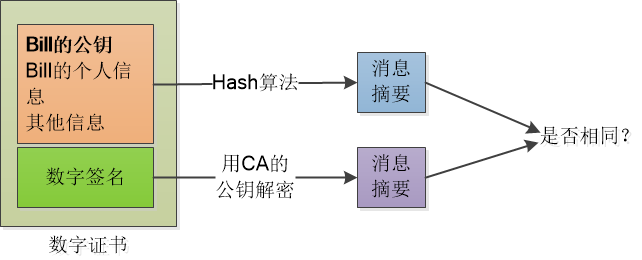
1. **利用CA的私钥对消息摘要进行加密生成签名。**



1. **再把原始信息和数据签名合并， 形成一个全新的东西，叫做“数字证书”**



1. **当Bill把他的证书发给张大牛的时候， 张大牛就用同样的Hash 算法， 再次生成消息摘要，然后用CA的公钥对数字签名解密， 得到CA创建的消息摘要， 两者一比，就知道有没有人篡改了！**



1. **如果没人篡改， 张大牛就可以安全的拿到Bill的公钥，有了公钥， 后序的加密工作就可以开始了。**

不过这些都是在必须得信任ＣＡ，并且通过安全的的方式获取他们的公钥的前提下，这样才能把游戏玩下去。

注：这些ＣＡ本身也有证书来证明自己的身份，并且ＣＡ的信用是像树一样分级的，高层的ＣＡ给底层的ＣＡ做信用背书，而操作系统／浏览器中会内置一些顶层的ＣＡ的证书，相当于你自动信任了他们。　这些顶层的ＣＡ证书一定得安全地放入操作系统／浏览器当中，否则世界大乱。

## 具体的Https流程

