https

问题：http是明文传输的，所以并不安全。

解决方案：使用ssl或者tls商量出一个密钥用作沟通。该密钥为对称加密密钥。

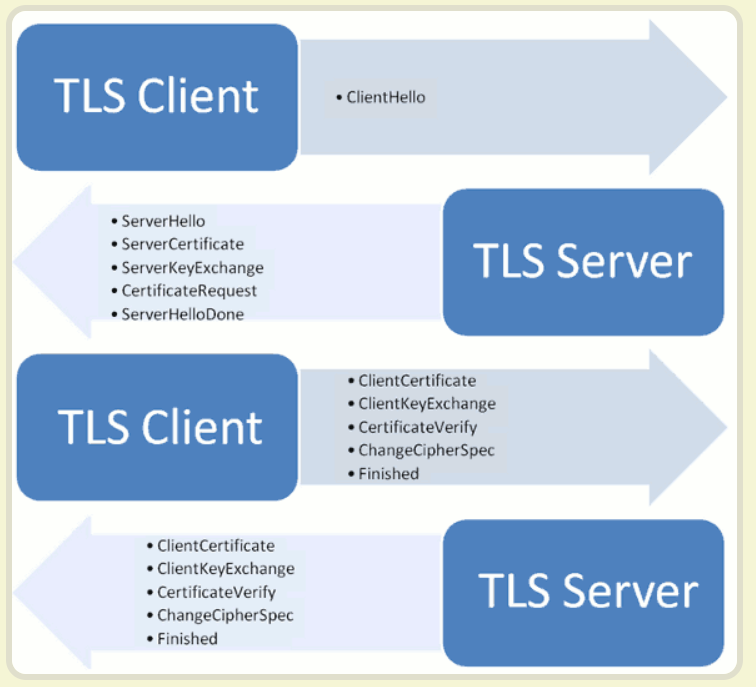
世界上原来没有坏人，大家通信都是通过http明文通信，并不会担心自己的消息被窃听，篡改或者干脆自己就被冒充了。

然而有一天，有一个人变成了坏蛋。他想，既然http传输的是明文，那么假如自己劫持到这个明文，就可以获得一些他人的隐私信息。假如自己再对这个明文修改一下，那么就可以告诉他人一些错误的信息。

为了防止这种情况的出现，善良的百姓设计了一种通信机制，来使得通信免受窃听、篡改和冒充的风险。这种机制就是https。

https的意思是，它将http上添加了一个ssl机制，使得通信变得隐秘。具体来说，就是服务器和客户端共享一个其他任何人都不知道的密钥。双方通信就用这个密钥来将自己的整个http报文加密，另一方收到密文后用同样的密钥进行解密（因为是对称加密）。这样双方就可以进行加密通话，别人并不知道他们在说什么。

那么问题来了，这个双方共享的别人都不知道的密钥究竟怎么形成的呢？肯定不可能直接告诉对方，因为那样万一被坏蛋截取了会话密钥，那一切不都白搭了？于是有人提出了一个方案，那就是ssl的四次握手。



1. 客户端发送请求（包括支持加密方式，协议版本，一个客户端生成的随机数用于生成对话密钥）
2. 服务端返回信息（服务端证书，确认加密通信协议版本，服务端生成的随机数用于生成对话密钥，确认加密方法）
3. 客户端返回信息（一个随机数（用在证书中的服务端的公钥加密），编码改变通知（宣布之后信息用加密通话），客户端握手结束通知）
4. 服务端返回信息（编码改变通知（宣布之后信息用加密通话），客户端握手结束通知）

这样，对话密钥就被双方共享了，而且除了双方之外别人都不知道。

之前提到了证书，这个东西是干什么用的呢。证书就是做身份验证用的。服务器要证明“我是我“，必须用证书表明身份。

设想一个问题，此时有一个攻击者谎称自己是google.com，然后给大家发了自己的公钥，那么怎么样才能知道这个人不是google.com呢？

方法就是，找一个CA（certificated authority）为自己的公钥颁发一个证书。这个证书的组成是 。然后证书中有着服务端的信息，比如公钥和服务端的名称，证书颁发的时间和失效时间等。

当客户端收到一个证书，首先用CA的公钥解密，证明该证书的确是由CA颁发的，然后查看里面的信息，认证了服务端的信息。

总的来说，就是A想给B发个消息，那就先用B的公钥加密（保密性），然后再用A的私钥加密（签名，sender authentication），最后再给这个密文盖个章（加上CA证书），完美。

证书文件 crt 密钥文件 pem