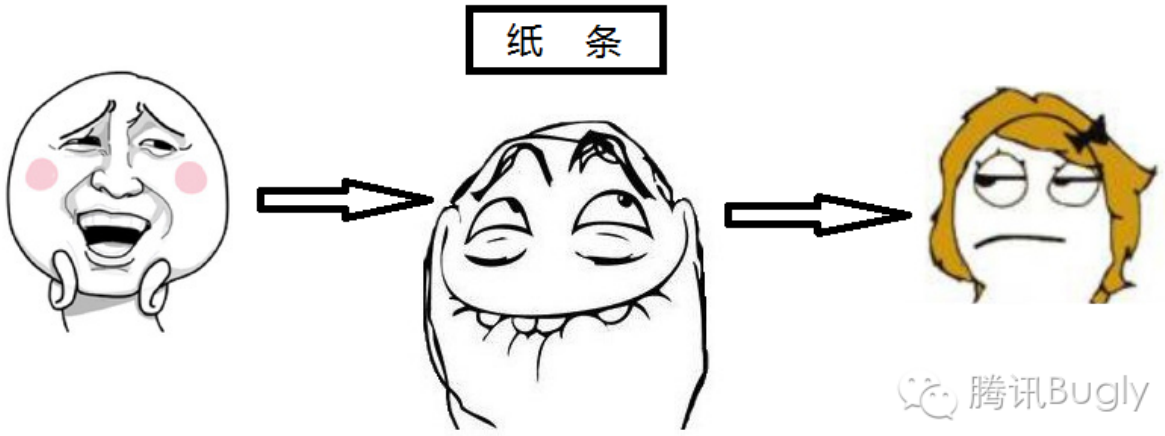
“HTTPS”安全在哪里？ - 技术分享 - Bugly

背景 最近基于兴趣学学习了下 HTTPS 相关的知识，在此记录下学习心得。 在上网获取信息的过程中，我们接触最多的信息加密传输方式也莫过于 HTTPS 了。每当访问一个站点，浏览器的地址栏中出现绿色图标时，意味着该站点支持 HTTPS 信息传输方式。我们知道 HTTPS 是我们常见的 HTTP 协议与某个加密协议的混合体，也就是 HTTP+S。这个 S 可以是 TLS（安全传输层协议）、也可以是 SSL（安全套接层），不过我更认可另一个抽象概括的说法，HTTP+Security。不过要谈论 HTTPS 为何安全，还得从 HTTP 为何不安全说起。 假设你现在正坐在教室里上课，现在你非常想和走道旁的迷人的 TA 说一些话，一般这个时候你会用“传纸条”的方式来交流。而这个方式和 TCP/IP 协议基本的工作模式十分相像：

- 通过小动作引起对方注意；
- 对方以多种可能的方式（注视、肢体语言等）回应于你；
- 你确认对方感知到你后，将纸条传给对方；
- 对方阅读纸条；
- 对方给予你阅读后的反应；

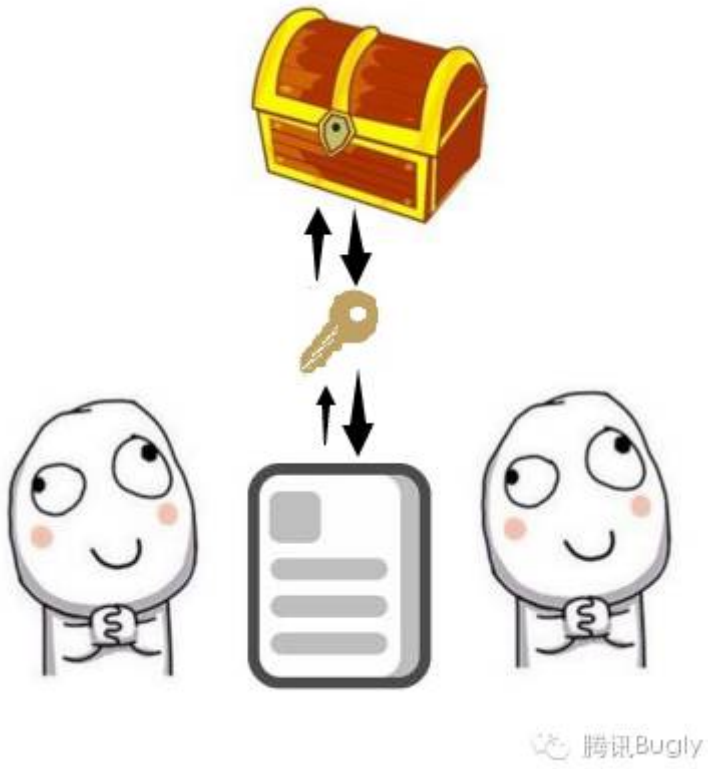
怎么样，这个流程是不是很熟悉？如果你要传递纸条的 TA 距离你很远怎么办？HTTP 协议就是指你在纸条上写明你要传给的 TA 是谁，或者 TA 的座位在哪，接着只需要途径的同学拿到纸条后根据纸条上的指示依次将纸条传过去就 OK 了。



这个时候问题来了：途径的同学完全可以观看并知道你在纸条上写了什么。

这就是 HTTP 传输所面临的问题之一：中间人攻击，指消息传递的过程中，处在传递路径上的攻击者可以嗅探或者窃听传输数据的内容。加密HTTPS 针对这个问题，采用了“加密”的方式来

解决。最著名原始的加密方法就是对称加密算法了，就是双方约定一个暗号，用什么字母替换什么字母之类的。现在一般采用一种叫 AES（高级加密算法）的对称算法。



对称加密算法既指加密和解密需要使用的密钥 key 是一样的。AES 在数学上保证了，只要你使用的 key 足够长，破解几乎是不可能的（除非光子计算机造出来了）我们先假设在没有密钥 key 的情况下，密文是无法被破解的，然后再回到这个教室。你将用 AES 加密后的内容噌噌噌地写在了纸条上，正要传出去的时候你突然想到，TA 没有 key 怎么解密内容呀，或者说，应该怎么把 key 给TA？如果把 key 也写在纸条上，那么中间人照样可以破解窃听纸条内容。也许在现实环境中你有其他办法可以把 key 通过某种安全的渠道送到 TA 的手里，但是互联网上的实现难度就比较大了，毕竟不管怎样，数据都要经过那些路由。于是聪明的人类发明了另一种加密算法——非对称加密算法。这种加密算法会生成两个密钥（key1 和 key2）。凡是 key1 加密的数据，key1 自身不能解密，需要 key2 才能解密；凡事 key2 加密的数据，key2 自身不能解密，只有 key1 才能解密。

> 目前这种算法有很多中，最常用的是 RSA。其基于的数学原理是：两个大素数的乘积很容易算，但是用这个乘积去算出是哪两个素数相乘就很复杂了。好在以目前的技术，分解大数的素因确实比较困难，尤其是当这个大数足够大的时候（通常使用2的10次方个二进制位那么大），就算是超级计算机，解密也需要非常长的时间。

现在就把这种非对称加密的方法应用在我们教室传纸条的场景里。

- 你在写纸条内容之前先用 RSA 技术生成了一对密钥 k1 和 k2。
- 你把 k1 用明文传了出去，路经也许有人会截取，但是没有用，k1 加密的数据需要 k2 才可以破解，而 k2 在你自己手中。
- k1 传到了目的人，目的人会去准备一个接下来准备用于对称加密（AES）的传输密钥 key，然后用收到的 k1 把 key 加密，传给你。

- 你用手上的 k2 解出 key 后，全教室只有你和你的目的人拥有这个对称加密的 key，你们俩就可以尽情聊天不怕窃听啦~

这里也许你会有问题，为什么不直接用非对称加密来加密信息，而是加密 AES 的 key 呢？
因为非对称加密和解密的平均消耗时间比较长，为了节省时间提高效率，我们通常只是用它来交换密钥，而非直接传输数据。然而使用非对称加密真的可以防范中间人攻击吗？
虽然看上去很安全，但是实际上却挡不住可恶的中间人攻击。

假设你是 A，你的目的地是 B，现在要途径一个恶意同学M。

中间人的恶意之处在于它会伪装成你的目标。

- 当你要和 B 完成第一次密钥交换的时候，M 把纸条扣了下来，假装自己是B并伪造了一个 key，然后用你发来的 k1 加密了 key 发还给你。
- 你以为你和 B 完成了密钥交换，实际上你是和 M 完成了密钥交换。
- 同事 M 和 B 完成一次密钥交换，让 B 以为和 A 你完成了密钥交换。
- 现在整体的加密流程变成了A（加密链接1）->M(明文)->B(加密链接2)的情况了，这时候 M 依然可以知道A和B传输的全部消息。

这个时候就是体现 HTTPS 和传纸条的区别了。在教室里，你是和一位与你身份几乎对等的的对象来通信；而在访问网站时，对方往往是一个比较大（或者知名）的服务者，他们有充沛的资源，或许他们可以向你证明他们的合法性。此时我们需要引入一个非常权威的第三方，一个专门用来认证网站合法性的组织，可以叫做 CA（Certificate Authority）。各个网站服务商可以向 CA 申请证书，使得他们在建立安全连接时可以带上 CA 的签名。而 CA 得安全性是由操作系统或者浏览器来认证的。你的 Windows、Mac、Linux、Chrome、Safari 等会在安装的时候带上一个他们

认为安全的 CA 证书列表，只有和你建立安全连接的网站带有这些CA的签名，操作系统和浏览器才会认为这个链接是安全的，否则就有可能遭到中间人攻击。一旦某个 CA 颁发的证书被用于的非法途径，那么这个 CA 之前颁发过的所有证书都将被视为不安全的，这让所有 CA 在颁发证书时都十分小心，所以 CA 证书在通常情况下是值得信任的。

总结使 HTTP 后面增加一个S（Security）的技术，正是 对称加密 + 非对称加密 + CA 认证这三种技术的混合体。当然这个主要是 HTTPS 的基本原理，真正实际中的 HTTPS 的协议是比以上的描述更为复杂一些的，并且其中任何一步稍有闪失，整个流程都将不再安全。这也是为什么 HTTPS 协议从 SSL 1.0升级到 SSL 3.0，再被 TLS 1.0 现在被 TLS 1.3取代，其背后都是一个个细节上的优化，以防有任何闪失。TLS 协议相比 SSL 协议增加了传输层的安全保证。

如果你觉得内容意犹未尽，如果你想了解更多相关信息，请扫描以下二维码，关注我们的公众账号，可以获取更多技术类干货，还有精彩活动与你分享~

54