梗概

HTTP：是互联网上应用最为广泛的一种网络协议，是一个客户端和服务器端请求和应答的标准（HTTP 协议运行在 TCP 之上），用于从 WWW 服务器传输超文本到本地浏览器的传输协议，它可以使浏览器更加高效，使网络传输减少。
HTTPS：是以安全为目标的 HTTP 通道，简单讲是 HTTP 的安全版，即 HTTP 下加入 SSL 层，HTTPS 的安全基础是 SSL，因此加密的详细内容就需要 SSL。HTTPS 协议的主要作用可以分为两种：一种是建立一个信息安全通道，来保证数据传输的安全；另一种就是确认网站的真实性。

区别

Http 协议运行在 TCP 之上，明文传输，客户端与服务器端都无法验证对方的身份；HTTPS 是运行在 SSL/TLS 之上的 HTTP 协议，SSL/TLS 运行在 TCP 之上。HTTPS 是添加了加密和认证机制的 HTTP。二者之间存在如下不同：
端口不同：Http 与 Http 使用不同的连接方式，用的端口也不一样，前者是 80，后者是 443；
资源消耗：和 HTTP 通信相比，Https 通信会由于加减密处理消耗更多的 CPU 和内存资源；
开销：Https 通信需要证书，而证书一般需要向认证机构购买；
Https 的加密机制是一种共享密钥加密和公开密钥加密并用的混合加密机制。
HTTPS 采用混合加密机制
由于公有密钥的机制相对复杂，导致其处理速度相对较慢。于是 HTTPS 利用了两者的优势，采用了混合加密的机制。我们知道，共享（对称）密钥未能解决的问题是如何能够安全地把密钥发送给对方。只要解决了这个问题就可以进行安全地通信。于是，HTTPS 首先是通过公有密钥来对共享密钥进行加密传输。当共享密钥安全地传输给对方后，双方则使用共享密钥的方式来加密报文，以此来提高传输的效率。

步骤 1：向服务器发起请求。
步骤 2-3：取出公有密钥及证书并发送给客户端。
步骤 4：客户端判断公有密钥是否有效，无效则显示警告。有效则生成一个随机数串，并以此生成客户端的共享密钥。
步骤 5：用步骤 3 得到的公有密钥对该随机数串加密，发送到服务器。
步骤 6：服务器得到加密报文，用私有密钥解密报文，得到随机数串，并以此生成服务器端的共享密钥。此时客户端和服务端拥有相同的共享密钥，可以用该共享密钥进行安全通信。
步骤 7-8：服务器对响应进行加密，客户端对报文进行解密。

HTTPS 的优点

尽管 HTTPS 并非绝对安全，掌握根证书的机构、掌握加密算法的组织同样可以进行中间人形式的攻击，但 HTTPS 仍是现行架构下最安全的解决方案，主要有以下几个好处：
（1）使用 HTTPS 协议可认证用户和服务器，确保数据发送到正确的客户机和服务器；

（2）HTTPS 协议是由 SSL+HTTP 协议构建的可进行加密传输、身份认证的网络协议，要比 http 协议安全，可防止数据在传输过程中不被窃取、改变，确保数据的完整性。
（3）HTTPS 是现行架构下最安全的解决方案，虽然不是绝对安全，但它大幅增加了中间人攻击的成本。

HTTPS 的缺点

虽然说 HTTPS 有很大的优势，但其相对来说，还是存在不足之处的：
（1）HTTPS 协议握手阶段比较费时，会使页面的加载时间延长近 50%，增加 10% 到 20% 的耗电；
（2）HTTPS 连接缓存不如 HTTP 高效，会增加数据开销和功耗，甚至已有的安全措施也会因此而受到影响；
（3）SSL 证书需要钱，功能越强大的证书费用越高，个人网站、小网站没有必要一般不会用。
（4）SSL 证书通常需要绑定 IP，不能在同一 IP 上绑定多个域名，IPv4 资源不可能支撑这个消耗。
（5）HTTPS 协议的加密范围也比较有限，在黑客攻击、拒绝服务攻击、服务器劫持等方面几乎起不到什么作用。最关键的，SSL 证书的信用链体系并不安全，特别是在某些国家可以控制 CA 根证书的情况下，中间人攻击一样可行。

HTTP 是不保存状态的协议，如何保存用户状态？

HTTP 是一种不保存状态，即无状态（stateless）协议。也就是说 HTTP 协议自身不对请求和响应之间的通信状态进行保存。那么我们保存用户状态呢？Session 机制的存在就是为了解决这个问题，Session 的主要作用就是通过服务端记录用户的状态。典型的场景是购物车，当你要添加商品到购物车的时候，系统不知道是哪个用户操作的，因为 HTTP 协议是无状态的。服务端给特定的用户创建特定的 Session 之后就可以标识这个用户并且跟踪这个用户了（一般情况下，服务器会在一定时间内保存这个 Session，过了时间限制，就会销毁这个 Session）。

在服务端保存 Session 的方法很多，最常用的就是内存和数据库 (比如是使用内存数据库 redis 保存)。既然 Session 存放在服务器端，那么我们如何实现 Session 跟踪呢？大部分情况下，我们都是通过在 Cookie 中附加一个 Session ID 来方式来跟踪。