**1、Http与Https的区别有哪些？**

（1）HTTP 的URL 以http:// 开头，而HTTPS 的URL 以https:// 开头；

（2）HTTP 是不安全的，而 HTTPS 是安全的；

（3）HTTP 标准端口是80 ，而 HTTPS 的标准端口是443；

（4）在OSI 网络模型中，HTTP工作于应用层，而HTTPS 的安全传输机制工作在传输层；

（5）HTTP无法加密，而HTTPS 对传输的数据进行加密；

（6）HTTP无需证书，而HTTPS 需要CA机构wosign的颁发的SSL证书。

**2、什么是无状态协议?怎么解决Http协议无状态协议?**

HTTP协议是无状态协议。无状态是指协议对于事务处理没有记忆能力。缺少状态意味着如果后续处理需要前面的信息，则它必须重传，这样可能导致每次连接传送的数据量增大。 也就是说，当客户端一次HTTP请求完成以后，客户端再发送一次HTTP请求，HTTP并不知道当前客户端是一个“老用户”。

可以使用Cookie来解决无状态的问题，Cookie就相当于一个通行证，第一次访问的时候给客户端发送一个Cookie，当客户端再次来的时候，拿着Cookie(通行证)，那么服务器就知道这个是“老用户”。

**3、常用的HTTP方法有哪些?**

GET： 用于请求访问已经被URI（统一资源标识符）识别的资源，可以通过URL传参给服务器

POST：用于传输信息给服务器，主要功能与GET方法类似，但一般推荐使用POST方式。

PUT： 传输文件，报文主体中包含文件内容，保存到对应URI位置。

HEAD： 获得报文首部，与GET方法类似，只是不返回报文主体，一般用于验证URI是否有效。

DELETE：删除文件，与PUT方法相反，删除对应URI位置的文件。

OPTIONS：查询相应URI支持的HTTP方法。

**4、一次完整的HTTP请求所经历的7个步骤？**

（1）建立TCP连接

（2）Web浏览器向Web服务器发送请求行

（3）Web浏览器发送请求头

（4）Web服务器应答

（5）Web服务器发送应答头

（6）Web服务器向浏览器发送数据

（7）Web服务器关闭TCP连接

一般情况下，一旦Web服务器向浏览器发送了请求数据，它就要关闭TCP连接。

如果浏览器或者服务器在其头信息加入了这行代码（表示此连接为长连接）：

Connection:keep-alive

TCP连接在发送后将仍然保持打开状态，于是，浏览器可以继续通过相同的连接发送请求。保持连接节省了为每个请求建立新连接所需的时间，还节约了网络带宽。

**5、HTTP请求报文与响应报文格式**

请求报文包含四部分：

a、请求行：包含请求方法、URI、HTTP版本信息

b、请求首部字段

c、请求内容实体

d、空行

响应报文包含四部分：

a、状态行：包含HTTP版本、状态码、状态码的原因短语

b、响应首部字段

c、响应内容实体

d、空行

**常见的首部：**

通用首部字段（请求报文与响应报文都会使用的首部字段）

Date：创建报文时间

Connection：连接的管理

Cache-Control：缓存的控制

Transfer-Encoding：报文主体的传输编码方式

请求首部字段（请求报文会使用的首部字段）

Host：请求资源所在服务器

Accept：可处理的媒体类型

Accept-Charset：可接收的字符集

Accept-Encoding：可接受的内容编码

Accept-Language：可接受的自然语言

响应首部字段（响应报文会使用的首部字段）

Accept-Ranges：可接受的字节范围

Location：令客户端重新定向到的URI

Server：HTTP服务器的安装信息

实体首部字段（请求报文与响应报文的的实体部分使用的首部字段）

Allow：资源可支持的HTTP方法

Content-Type：实体主类的类型

Content-Encoding：实体主体适用的编码方式

Content-Language：实体主体的自然语言

Content-Length：实体主体的的字节数

Content-Range：实体主体的位置范围，一般用于发出部分请求时使用

**6、HTTPS工作原理？**

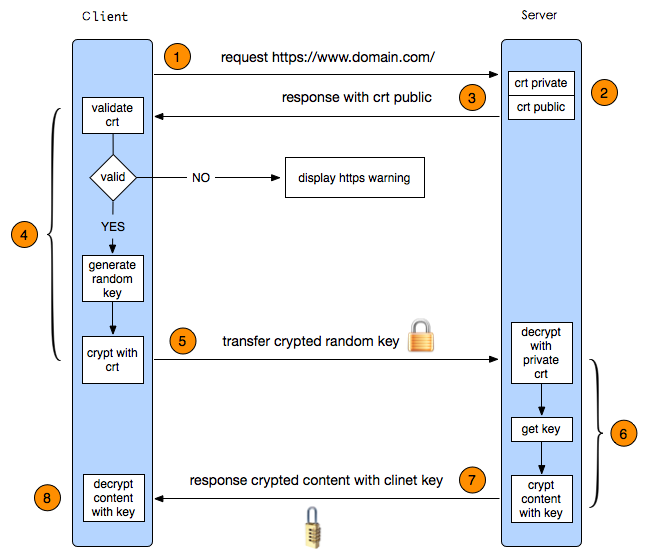
一、首先HTTP请求服务端生成证书，客户端对证书的有效期、合法性、域名是否与请求的域名一致、证书的公钥（RSA加密）等进行校验；

二、客户端如果校验通过后，就根据证书的公钥的有效， 生成随机数，随机数使用公钥进行加密（RSA加密）；

三、消息体产生的后，对它的摘要进行MD5（或者SHA1）算法加密，此时就得到了RSA签名；

四、发送给服务端，此时只有服务端（RSA私钥）能解密。

五、解密得到的随机数，再用AES加密，作为密钥（此时的密钥只有客户端和服务端知道）。



**7、HTTP优化方案**

（1）TCP复用：TCP连接复用是将多个客户端的HTTP请求复用到一个服务器端的TCP连接上，HTTP复用是指一个一个客户端的多个HTTP请求通过一个TCP连接进行处理。

（2）内容缓存：将经常用到的内容进行缓存起来，那么客户端就可以直接在内存中获取相应的数据了。

（3）压缩：将文本数据进行压缩，减少宽带。

（4）SSL加速：使用SSL协议对HTTP协议进行加密，在通道内加密并加速。

（5）TCP缓冲：通过TCP缓冲技术，可以提高服务器端响应事件和处理效率，减少由于通信链路问题给服务器造成连接负担。