### 阿里巴巴 你对HTTPS原理是怎么理解的

本专栏专注分享大型Bat面试知识，后续会持续更新，喜欢的话麻烦点击一个star

HTTPS（全称：HyperText Transfer Protocol over Secure Socket Layer），其实 HTTPS 并不是一个新鲜协议，Google 很早就开始启用了，初衷是为了保证数据安全。 近两年，Google、Baidu、Facebook 等这样的互联网巨头，不谋而合地开始大力推行 HTTPS， 国内外的大型互联网公司很多也都已经启用了全站 HTTPS，这也是未来互联网发展的趋势。

为鼓励全球网站的 HTTPS 实现，一些互联网公司都提出了自己的要求：

1）Google 已调整搜索引擎[算法](http://lib.csdn.net/base/datastructure)，让采用 HTTPS 的网站在搜索中排名更靠前；

2）从 2017 年开始，Chrome 浏览器已把采用 HTTP 协议的网站标记为不安全网站；

3）苹果要求 2017 年 App Store 中的所有应用都必须使用 HTTPS 加密连接；

4）当前国内炒的很火热的[微信](http://lib.csdn.net/base/wechat)小程序也要求必须使用 HTTPS 协议；

5）新一代的 HTTP/2 协议的支持需以 HTTPS 为基础。

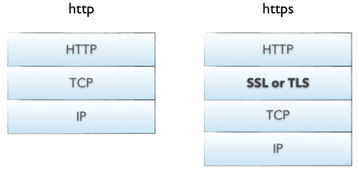
等等，因此想必在不久的将来，全网 HTTPS 势在必行。

### **概念**

#### **协议**

1、HTTP 协议（HyperText Transfer Protocol，超文本传输协议）：是客户端浏览器或其他程序与Web服务器之间的应用层通信协议 。

2、HTTPS 协议（HyperText Transfer Protocol over Secure Socket Layer）：可以理解为HTTP+SSL/TLS， 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，用于安全的 HTTP 数据传输。



img

如上图所示 HTTPS 相比 HTTP 多了一层 SSL/TLS

SSL（Secure Socket Layer，安全套接字层）：1994年为 Netscape 所研发，SSL 协议位于 TCP/IP 协议与各种应用层协议之间，为数据通讯提供安全支持。

TLS（Transport Layer Security，传输层安全）：其前身是 SSL，它最初的几个版本（SSL 1.0、SSL 2.0、SSL 3.0）由网景公司开发，1999年从 3.1 开始被 IETF 标准化并改名，发展至今已经有 TLS 1.0、TLS 1.1、TLS 1.2 三个版本。SSL3.0和TLS1.0由于存在安全漏洞，已经很少被使用到。TLS 1.3 改动会比较大，目前还在草案阶段，目前使用最广泛的是TLS 1.1、TLS 1.2。

#### **加密算法：**

据记载，公元前400年，古希腊人就发明了置换密码；在第二次世界大战期间，德国军方启用了“恩尼格玛”密码机，所以密码学在社会发展中有着广泛的用途。

1、对称加密

有流式、分组两种，加密和解密都是使用的同一个密钥。

例如：DES、AES-GCM、ChaCha20-Poly1305等

2、非对称加密

加密使用的密钥和解密使用的密钥是不相同的，分别称为：公钥、私钥，公钥和算法都是公开的，私钥是保密的。非对称加密算法性能较低，但是安全性超强，由于其加密特性，非对称加密算法能加密的数据长度也是有限的。

例如：RSA、DSA、ECDSA、 DH、ECDHE

3、哈希算法

将任意长度的信息转换为较短的固定长度的值，通常其长度要比信息小得多，且算法不可逆。

例如：MD5、SHA-1、SHA-2、SHA-256 等

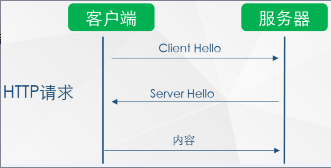
4、数字签名

签名就是在信息的后面再加上一段内容（信息经过hash后的值），可以证明信息没有被修改过。hash值一般都会加密后（也就是签名）再和信息一起发送，以保证这个hash值不被修改。

### **详解**

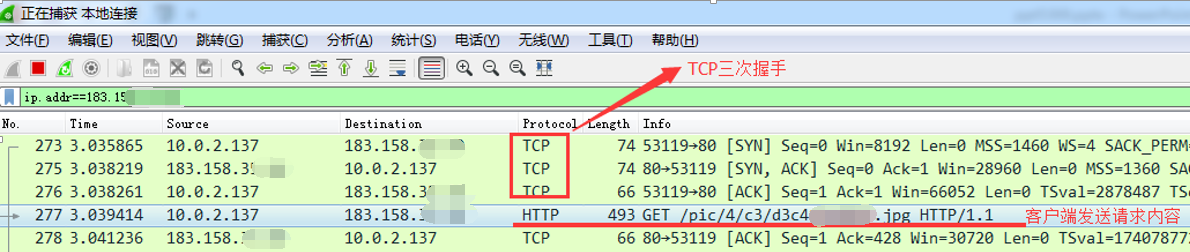
一、HTTP 访问过程

抓包如下：



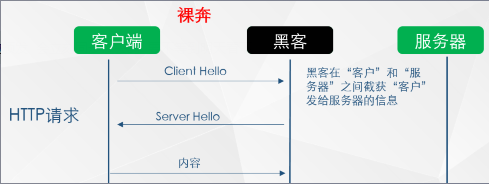
img

如上图所示，HTTP请求过程中，客户端与服务器之间没有任何身份确认的过程，数据全部明文传输，“裸奔”在互联网上，所以很容易遭到黑客的攻击，如下：



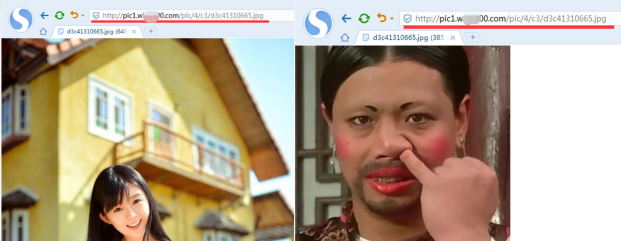
img

可以看到，客户端发出的请求很容易被黑客截获，如果此时黑客冒充服务器，则其可返回任意信息给客户端，而不被客户端察觉，所以我们经常会听到一词“劫持”，现象如下：



img

下面两图中，浏览器中填入的是相同的URL，左边是正确响应，而右边则是被劫持后的响应。



img

所以 HTTP 传输面临的风险有：

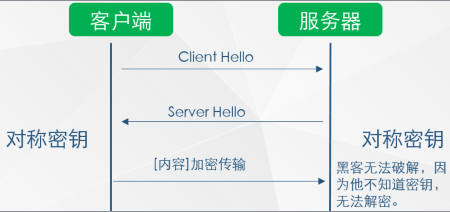
（1） 窃听风险：黑客可以获知通信内容。

（2） 篡改风险：黑客可以修改通信内容。

（3） 冒充风险：黑客可以冒充他人身份参与通信。

二、HTTP 向 HTTPS 演化的过程

第一步：为了防止上述现象的发生，人们想到一个办法：对传输的信息加密（即使黑客截获，也无法破解）



img

如上图所示，此种方式属于对称加密，双方拥有相同的密钥，信息得到安全传输，但此种方式的缺点是：

（1）不同的客户端、服务器数量庞大，所以双方都需要维护大量的密钥，维护成本很高

（2）因每个客户端、服务器的安全级别不同，密钥极易泄露

第二步：既然使用对称加密时，密钥维护这么繁琐，那我们就用非对称加密试试



img

如上图所示，客户端用公钥对请求内容加密，服务器使用私钥对内容解密，反之亦然，但上述过程也存在缺点：

（1）公钥是公开的（也就是黑客也会有公钥），所以第 ④ 步私钥加密的信息，如果被黑客截获，其可以使用公钥进行解密，获取其中的内容

第三步：非对称加密既然也有缺陷，那我们就将对称加密，非对称加密两者结合起来，取其精华、去其糟粕，发挥两者的各自的优势



img

如上图所示

（1）第 ③ 步时，客户端说：（咱们后续回话采用对称加密吧，这是对称加密的算法和对称密钥）这段话用公钥进行加密，然后传给服务器

（2）服务器收到信息后，用私钥解密，提取出对称加密算法和对称密钥后，服务器说：（好的）对称密钥加密

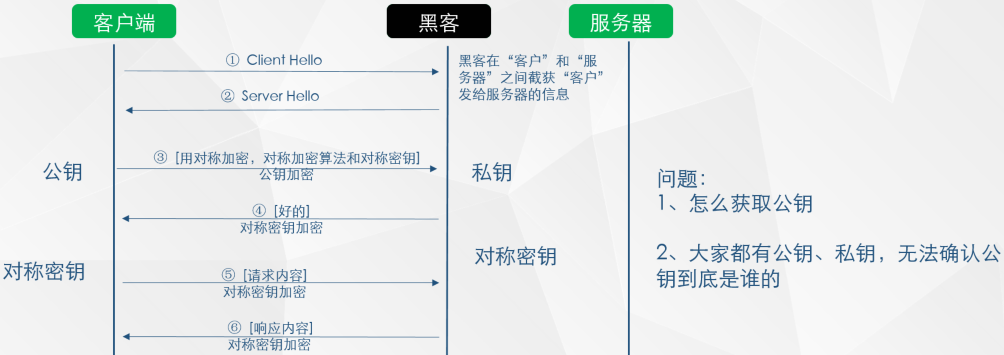
（3）后续两者之间信息的传输就可以使用对称加密的方式了

遇到的问题：

（1）客户端如何获得公钥

（2）如何确认服务器是真实的而不是黑客

第四步：获取公钥与确认服务器身份

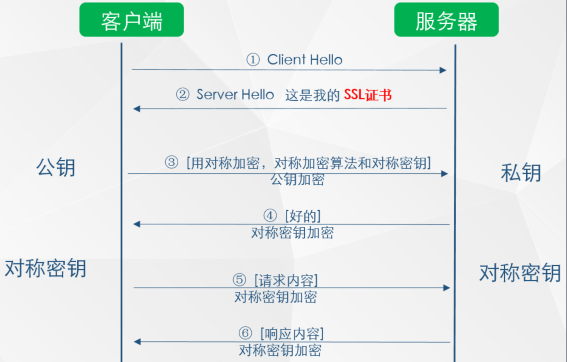


img

1、获取公钥

（1）提供一个下载公钥的地址，回话前让客户端去下载。（缺点：下载地址有可能是假的；客户端每次在回话前都先去下载公钥也很麻烦） （2）回话开始时，服务器把公钥发给客户端（缺点：黑客冒充服务器，发送给客户端假的公钥）

2、那有木有一种方式既可以安全的获取公钥，又能防止黑客冒充呢？ 那就需要用到终极武器了：SSL 证书



img

如上图所示，在第 ② 步时服务器发送了一个SSL证书给客户端，SSL 证书中包含的具体内容有：

（1）证书的发布机构CA

（2）证书的有效期

（3）公钥

（4）证书所有者

（5）签名

………

3、客户端在接受到服务端发来的SSL证书时，会对证书的真伪进行校验，以浏览器为例说明如下：

（1）首先浏览器读取证书中的证书所有者、有效期等信息进行一一校验

（2）浏览器开始查找[操作系统](http://lib.csdn.net/base/operatingsystem)中已内置的受信任的证书发布机构CA，与服务器发来的证书中的颁发者CA比对，用于校验证书是否为合法机构颁发

（3）如果找不到，浏览器就会报错，说明服务器发来的证书是不可信任的。

（4）如果找到，那么浏览器就会从操作系统中取出 颁发者CA 的公钥，然后对服务器发来的证书里面的签名进行解密

（5）浏览器使用相同的hash算法计算出服务器发来的证书的hash值，将这个计算的hash值与证书中签名做对比

（6）对比结果一致，则证明服务器发来的证书合法，没有被冒充

（7）此时浏览器就可以读取证书中的公钥，用于后续加密了

4、所以通过发送SSL证书的形式，既解决了公钥获取问题，又解决了黑客冒充问题，一箭双雕，HTTPS加密过程也就此形成

所以相比HTTP，HTTPS 传输更加安全

（1） 所有信息都是加密传播，黑客无法窃听。

（2） 具有校验机制，一旦被篡改，通信双方会立刻发现。

（3） 配备身份证书，防止身份被冒充。

### **总结**

综上所述，相比 HTTP 协议，HTTPS 协议增加了很多握手、加密解密等流程，虽然过程很复杂，但其可以保证数据传输的安全。所以在这个互联网膨胀的时代，其中隐藏着各种看不见的危机，为了保证数据的安全，维护网络稳定，建议大家多多推广HTTPS。

HTTPS 缺点：

（1）SSL 证书费用很高，以及其在服务器上的部署、更新维护非常繁琐

（2）HTTPS 降低用户访问速度（多次握手）

（3）网站改用HTTPS 以后，由HTTP 跳转到 HTTPS 的方式增加了用户访问耗时（多数网站采用302跳转）

（4）HTTPS 涉及到的安全算法会消耗 CPU 资源，需要增加大量机器（https访问过程需要加解密）