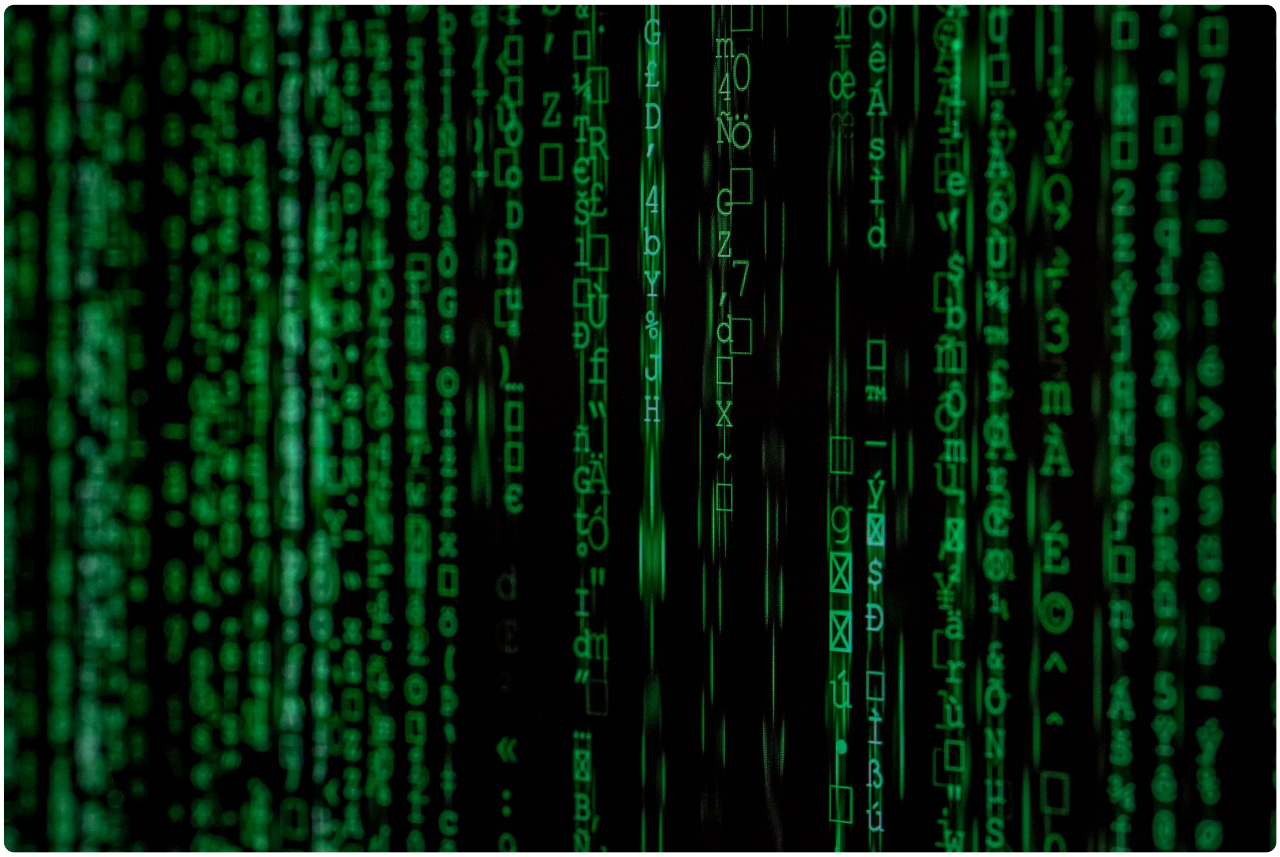
15 加密你的通话记录：从 HTTP 到 HTTPS

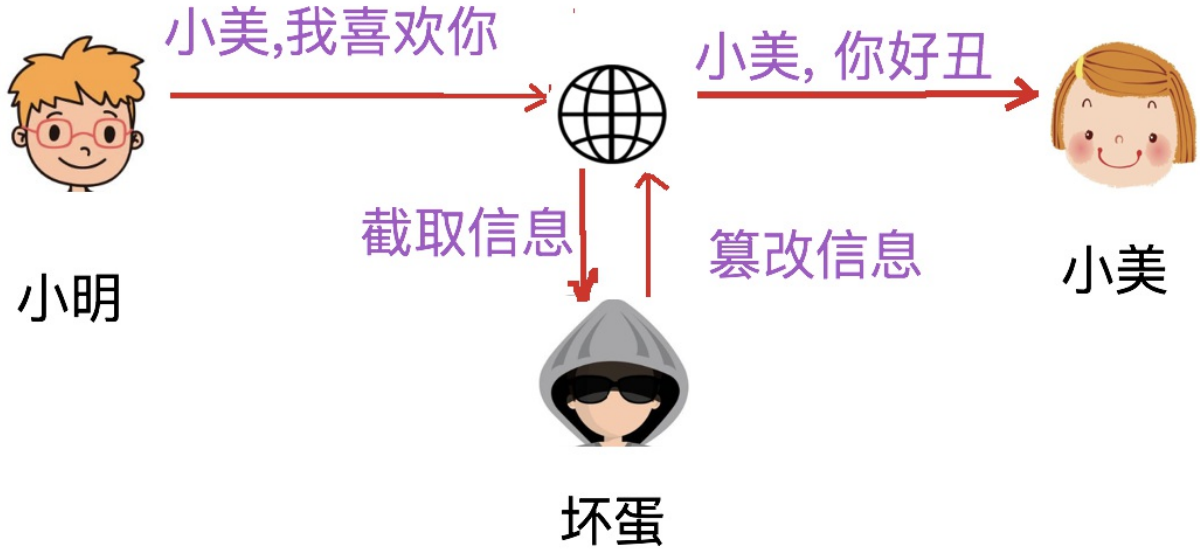
更新时间：2020-01-14 10:04:46



> 书是人类进步的阶梯。——高尔基

前言

大家可能都使用各种抓包软件进行过抓包操作吧，由于HTTP协议是非加密解析，也就是说我们通过HTTP发送的内容都是裸奔在网络中的，其他人可以清楚的看到我们HTTP报文内容了，这样的话，哪些坏人就能够修改我们的信息了。



为了防止这种情况，就出现了加密算法，HTTPS就是使用了加密算法的HTTP协议。
我们本篇文章就先给大家介绍一下加密算法相关内容。

加密算法

在学习加密算法之前优先了解一些和加密相关的概念。
明文：我们想要发送的真正的消息。
密文：对明文进行加密之后得到的消息，密文是真正要发送出去的内容。
密钥：顾名思义，所谓的密钥就是一种用于加密的钥匙，是我们加密所使用的规则。

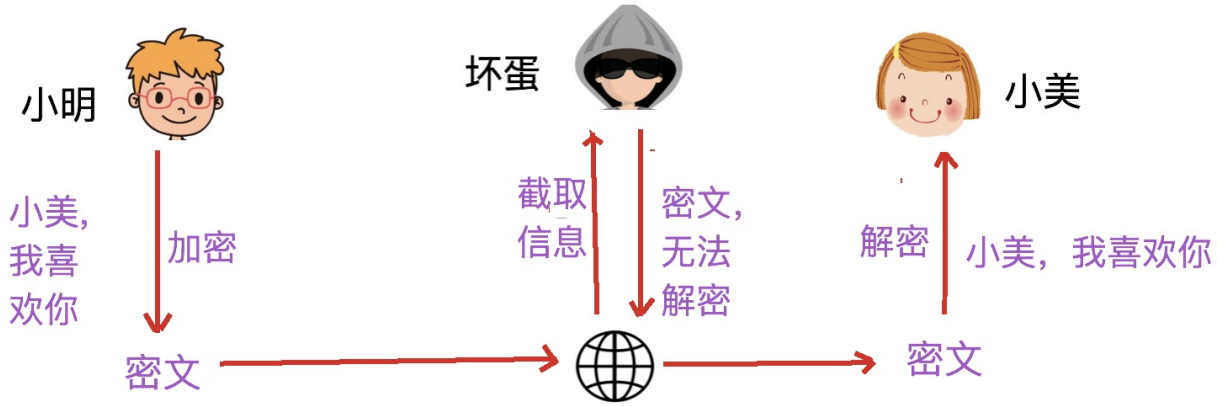
我记得很早之前看过一个谍战剧，其中有一个情节，一个特务要向另一个特殊传递一个消息，比如是小心李鬼，在传递消息的时候他会使用一长串的数字27 85 17 6 96 37 21 32。原来这些数字每两个为一组，每组的第一个数字是一本词典的页码，第二个数字是是当前页的第几个字。当另一个人收到这串数字之后会从根据这个规则进行反解密，之后就可以得到真正的信息了。

在这个情节中，小心李鬼就是明文，这一串数字就是密文，而这个加密规则就是密钥。

计算机世界中的加密算法演化了很多代，所以存在很多的加密的算法。我们熟知的加密方式有对称加密和非对称加密。

对称加密

对称加密就是通信的双方持有相同的密钥。小明使用密钥将要发送的内容进行加密，小美收到内容之后使用相同的密钥进行解密就可以了。上面特务传递消息例子使用的就是对称加密。



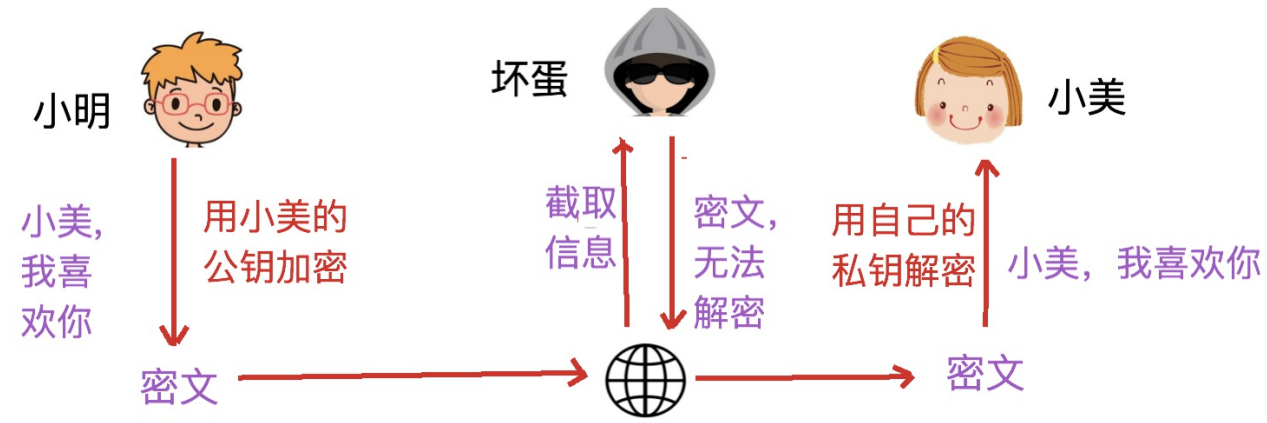
这种情况下我们传递的信息在一定程度上就保证了安全性，但是前提是密钥不能丢失，否则坏蛋还是能够修改我们的信息。（比如小明和小美在网络上互相交换密钥的时候被坏蛋截取了，那么坏蛋就知道密钥了）

非对称加密

大家都用过git吧，是否还记得初始化git的时候要生成一对密钥吗？对，这对密钥就是用于加密的。
非对称密钥包含了两个密钥，一个叫做公钥，一个叫做私钥。我们可以从自己的电脑的.ssh目录中看到这对密钥。

```
total 40
-rw-------@ 1 zhengxuyao  staff   216B  9 19 09:31 config
-rw-------  1 zhengxuyao  staff   1.6K  4 27  2017 id_rsa
-rw-r--r--  1 zhengxuyao  staff   401B  4 27  2017 id_rsa.pub
-rw-------  1 zhengxuyao  staff   3.4K 12 16 10:18 known_hosts
-rw-------  1 zhengxuyao  staff   2.8K  5 29  2019 known_hosts.old
```

这对密钥有什么特点呢？其实很简单，使用公钥加密之后的内容只能使用私钥进行解密，反之亦然。这样的话我们就可以把公钥告诉所有人了。别人把想要发送给你的信息使用你的公钥进行加密，你收到之后通过私钥进行解密就行了，这样即使坏蛋拿到了公钥也无法解密你的信息。因为此时我们只用把公钥发布出去，而私钥一直保存在本地，所以这种加密方式要比对称加密更加安全。



其实非对称加密还有一个非常重要的作用，就是身份认证。比如小美使用自己的私钥加密了一段信息发送给小明，如果小明收到之后可以用小美的公钥解密成功，那么就表示这段信息是小美发送的。这个功能就是加密过程中身份认证的含义。

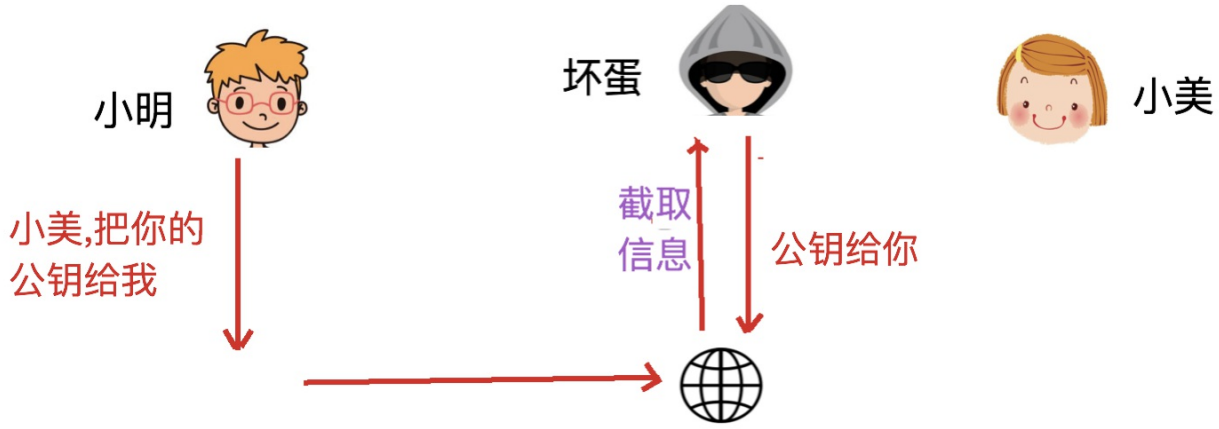
数字证书

我们上面说了对称加密， 非对称加密，这里面有一个很关键的因素没有说，通信双方如何把公钥发送给对方？我们可能会想到两种方法：

- 去对方的网站下载
- 每次通信的时候，由服务器把公钥发送给对方

第一种方法：我们不能确定下载的网址一定是服务器的，毕竟现在有很多钓鱼网站。并且，也不可能让用户在访问网站访问之前先去下载公钥(这也太麻烦了，并且还有很多非专业人士可能根本搞不懂什么是私钥,什么是公钥)。

第二种方法：这种方法是毕竟可行的，这样的话，用户在访问服务器的时候就可以无感知的下载公钥了（所有的操作都会自动完成，用户无需操作），但是这样也存在一个问题， 坏蛋可能劫持消息，然后把自己的公钥发送给了用户。



就像上图一样， 坏蛋截取了小明的信息，然后把自己的公钥发送给了小明。

那该如何防止这种情况呢？

数字证书就是为了防止这种情况。服务器可以向是证书颁发机构申请数字证书，这个数字证书包含了公钥，并且包含了申请者的详细信息，服务器可以把数字证书发送给用户，用户通过校验就可以知道这个数字证书是不是伪造的（校验过程是由操作系统完成的，每个操作系统都有默认的受信任证书机构，所以这样可以从操作系统级别防止坏蛋），这样就可以拿到公钥了。

数字证书很复杂，但是对于我么使用者来说，无需考虑那么多，我们只需要知道它是干什么的，仅此而已。

总结

本文我们介绍了一些加密相关的内容，这是理解HTTPS的一些基础，后面我们会分析如何使用HTTPS构建一个安全的服务器。

}

14 保留证据：能玩出花来的 Nginx 日志

16 HTTPS 当家做主