本文主要讲述了HTTPS的基本原理，通过HTTPS握手过程、证书链、中间人攻击、CA机构等问题，详细解释了百付宝系统中用到的HTTPS安全知识，同时，介绍了如何查看www.baifubao.com的线上证书及其含义。最后藉由查看证书，来屏蔽不安全证书带来的潜在风险。

Icon

“互联网仍然处于开端的开端阶段（the beginning of its beginning）”《失控》的作者凯文.凯利如是说。

仅仅是这开端的几十年就已经创造出了比过去所有的历史创造的财富的总和还要多。

如果信息不安全，那么互联网创造的财富不能被保证。HTTPS为此而生。

**1. HTTPS概述**

        HTTPS并不是一个单独的协议，而是对工作在一加密连接（SSL/TLS）上的常规HTTP协议。通过在TCP和HTTP之间加入**TLS(Transport Layer Security)**来加密.

(注: TLS为新版本的SSL)



**2. 为何需要HTTPS？**

        不使用SSL/TLS的HTTP通信，就是不加密的通信。所有信息明文传播，带来了三大风险。

Icon

* **窃听风险**（eavesdropping）：第三方可以获知通信内容。
* **篡改风险**（tampering）：第三方可以修改通信内容。
* **冒充风险**（pretending）：第三方可以冒充他人身份参与通信。

        SSL/TLS协议是为了解决这三大风险而设计的。那么SSL/TLS如何解决这三大风险呢？

**2.1 所有信息都是加密传播，第三方无法窃听**

        HTTPS一般使用的加密与HASH算法如下：

Icon

**非对称加密算法**：RSA，DSA/DSS

**对称加密算法**：AES，RC4，3DES

**HASH算法**：MD5，SHA1，SHA256

        这里就要比较对称加密和非对称加密了。

* **对称加密**:

Icon

加密和解密的密钥是**相同的**.

假设A和B之间通信，使用对称密钥是123.

1. A和B写信需要用123加密.
2. B收到A的信息，同样需要123解密。

**对称加密的最大风险**在于密钥一旦泄露，那么被截获的信件内容就会被破解.

* **非对称加密:**

Icon

加密和解密的密钥是**不同的**分为**私钥和公钥**。

私钥: 只有一份, 保存在收信人手中。不会在通信中传输，不会被泄露。

公钥: 可以有多份，保存在写信人手中。

假设B要给A写信:

1. A先生成一对公钥(123)和私钥(456)
2. A要把自己的公钥(456)通过某种安全的方式(数字证书，下文会提到)交给B
3. B用公钥(456)加密信件内容然后发给A
4. A使用私钥(123)解开内容

因此即使Step2时公钥泄露，那么即使B给A写的信被截获，由于没有A的私钥依然无法被解开.

**2.2 具有校验机制，一旦被篡改，通信双方会立刻发现**

Icon

使用HASH算法进行校验内容是否被篡改.

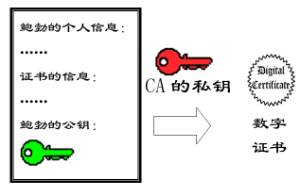
**2.3 配备数字证书，防止身份被冒充**

Icon

如果数字证书是CA认证的，那么就可以避免身份冒充.

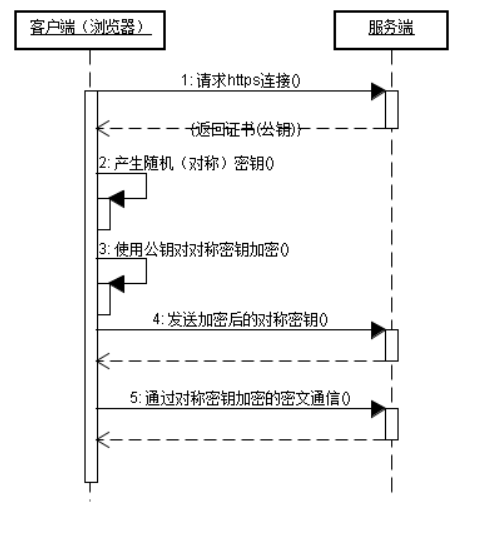
什么是数字证书呢？ 何为CA？

        数字证书是一个文件。此文件保存了加密过的用户的信息及公钥。对于百度钱包而言，数字证书就包含经过CA认证(使用CA的私钥加密)的域名（www.baifubao.com[）以及公钥等信息。](http://baifubao.com/)



        数字证书在HTTPS的什么时候会用上呢？这里就要提到HTTPS握手。

**3. HTTPS握手过程**



Icon

* Step1: 服务端返回的数字证书(此为用户证书)，客户端（浏览器）会进行如下验证:  
  1)   遍历计算机以及浏览器中保存的根证书（ROOT CA）和中间证书（Intermediate CA），若其中某个根证书或中间证书的**公钥**可以解开server端的证书，获得server的公钥和server域名. 否则握手失败。 **整个HTTPS通信的唯一核心保障就是可信的根证书。**  
  2)   判断证书中的域名（www.baifubao.com[）是否和正在访问的域名相同。否则认证失败，浏览器会提示证书不可信。但是握手加密依然可以继续。](http://baifubao.com/)
* Step2: 客户端(浏览器)产生一个随机的**对称密钥**
* Step3: 使用证书中服务端的**公钥加密**此**对称密钥**
* Step4: 将此已经加密过的对称密钥发给server。至此client和server都通过可靠的手段拥有对称密钥.
* Step5: 通过**对称密钥**加密Http的通信内容。

        通过上述的握手过程可以直到，**非对称密钥**只是用来**加密传输对称密钥**的。因为可以保证传输中的对称密钥在传输的过程中无法被破解。所以握手之后的内容通信就是安全的。

        握手过程中涉及到两种证书：

Icon

* https握手时server发给client的证书成为用户证书
* client所在计算机中原本保存的是 根证书（ROOT CA）和中间证书（Intermediate CA）

**为何需要可信的根证书呢？这就需要提到中间人攻击**

**4. 中间人攻击**

        假设A是服务器，B是用户，B向A发起HTTPS连接，于是A需要将自己的公钥发给B。中间人C通过某种手段可以截获并伪造AB之间的通讯（比如GFW或者共享wifi等）。

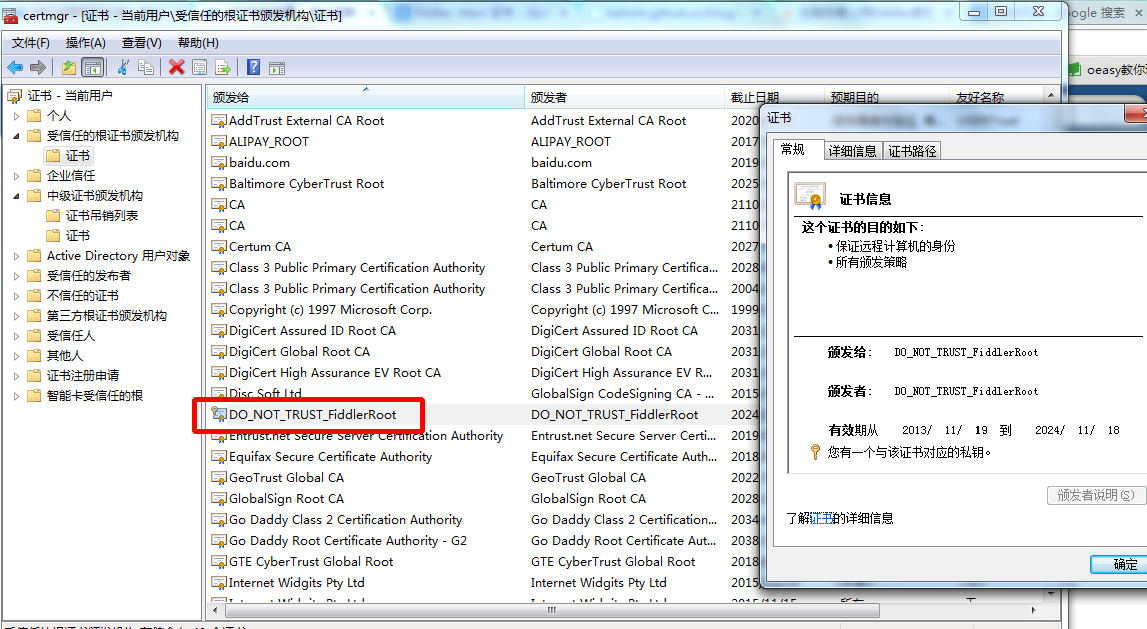
        那么C可以伪造一份A的公钥，并保有这分假公钥的私钥；然后截获A发给B的真公钥，并伪造自己的身份，让B认为自己就是A，并把伪造的公钥发给B；如此一来，B会通过伪造的公钥给A发送密文，而C就可以截获这些密文并利用手中的私钥轻易的解密这些密文了；然后将这些密文通过正确的公钥转发给服务器A，这样AB之间的通讯仍将继续，AB在毫不知情的情况下被中间人把证书“偷梁换柱”，从而达成了中间人攻击，AB之间的非对称加密形同虚设，从而TLS协议的对称加密的密钥就能被C轻易的获取。

        如此，TLS完全告破。

**总结：如果你安装了 中间人（代理服务器）自己生成的根证书，那么就中招了。**

**举个例子：**

        Fiddler对https的抓取就是靠中间人攻击的方式。打开Fiddler ,Tools->Fiddler options->HTTPS中的https capture，那么fidder会提示你安装一个fiddler自己生成的根证书(fiddler自己作为CA)名为DO\_NOT\_TRUST\_FIDDLER\_ROOT的根证书。如图:



**之后当浏览器访问的每个https的域名(如www.baifubao.com**[**等)：**](http://google.com/)

Icon

* fiddler会作为客户端先解开https内容
* 使用自己作为CA(名为DO\_NOT\_TRUST\_FIDDLER\_ROOT)的私钥，对baifubao这个域名颁发一个用户证书。
* https握手时向浏览器发送这个用户证书



**浏览器收到fiddler作为server端返回的https回应时:**

Icon

* 尝试解开server发来的用户证书，由于已经安装了fiddler的根证书，因此可以解开。
* 其中的域名是[google.com](http://google.com/),同浏览器访问的域名一致，因此证书这一部分就验证成功了。接下来的握手就自然可以完成。

        CA机构到底是什么呢？

**5. CA，证书链，根证书**

**5.1 数字证书认证机构（CA）**

* 它的出现就是为了防止中间人攻击的。
* 防止中间人攻击，说白了就是要确保B收到的A的公钥（证书）真实有效，这样数字证书认证机构应运而生。
* 数字证书认证机构说白了就相当于一个受信任的中间人。CA有一对根密钥，其公钥称为根证书。

Icon

A向CA申请一个证书，则CA利用其私钥加密A的公钥，其结果就是“服务器A，通过CA验证的证书”。

而在用户的操作系统（或者浏览器）中，会集成世界范围内所有被信任的CA的根证书。

这样，用户B在收到A发送给他的证书后，需要利用CA的根证书（公钥）解密后才能得到正确的公钥，如此一来，就完成了对A发送过来的信息的验证，证明了A的正身，不是C伪造的假证书，从而达成了中间人攻击的防范。

**5.2 数字证书从何而来**

**5.2.1 对于根证书（ROOT CA）和中间证书（Intermediate CA）**

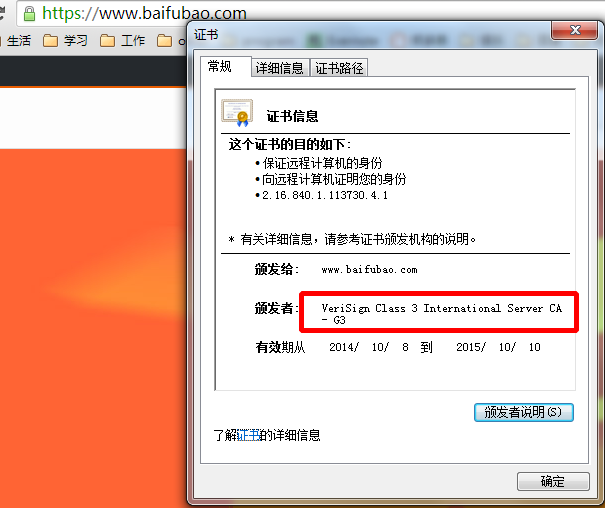
* 在用户的操作系统（或者浏览器）中，会集成世界范围内所有被信任的CA的根证书。
* 安装软件时顺带安装，如支付宝安全控件等，此时软件会请求管理员权限（流氓...）。
* 用户自己下载安装,如12306网站

**5.2.2 对于用户证书**

        比如www.baifubao.com[的证书，是百付宝公司通过如versign之类相关认证机构去资质审核以及缴费获得的。那么https访问中，server发来的数字证书长啥样呢？](http://baifubao.com/)

**5.3 百度钱包www.baifubao.com的证书**

        点击浏览器导航栏左上角的小锁





        www.baifubao.com证书路径中有三层，表示三级证书链。那么什么是证书链呢？

**5.4 证书链**

        CA证书分为两类：**根证书（Root CA）和中间证书（Intermediate CA）**。但是根证书的使用是收到严格限制的，不可能对于每一类用户都使用根证书去签发子数字证书，所以就有了中间证书的概念。

**中间证书**由根证书或上一级中间证书签发，它可以再往下级签发数字证书。

例如我们自己为某个域名申请了证书 My CA，那么对于三级证书链，它的签发过程如下：

Icon

Root CA 签发 Intermediate CA， Intermediate CA 签发 My CA这时我们就可以用My CA去给域名作数字认证了。

上面讲到的签发关系很像链式结构，所以被称作**证书链**。

**验证的过程**可想而知，就是签发的逆过程，这是通过证书链来完成的:

Icon

浏览器会在计算机以及浏览器的证书列表中查找此CA是否可信, 如果有则认为My CA是可信的；

如果没有，继续往上找，直到根证书：

如果根证书是可信的，那么整条证书链就是可信的；

如果根证书不可信，那么My CA将被认作是不可信的，浏览器就会发出警告。

所以说，对于刚才www.baifubao.com[的3级证书链来说，根证书(ROOT CA)是Versign Class3 Public xxxx . 中间证书(Intermediate CA)是Versign Class3 International xxxx](http://baifubao.com/)

那么如何查看自己安装的根证书呢？

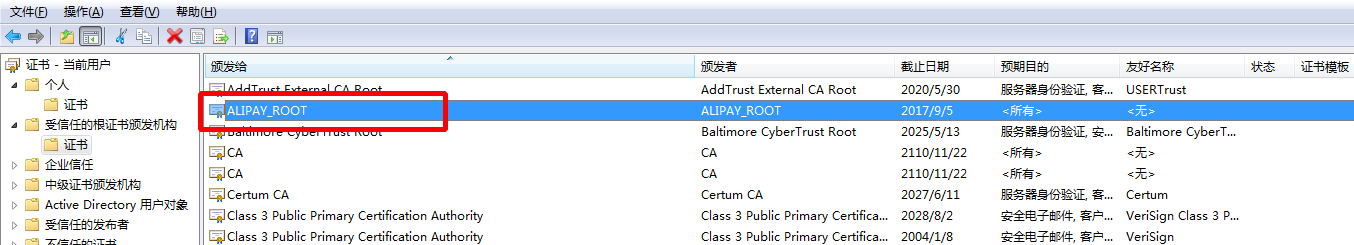
**5.5 如何查看已经安装的数字证书**

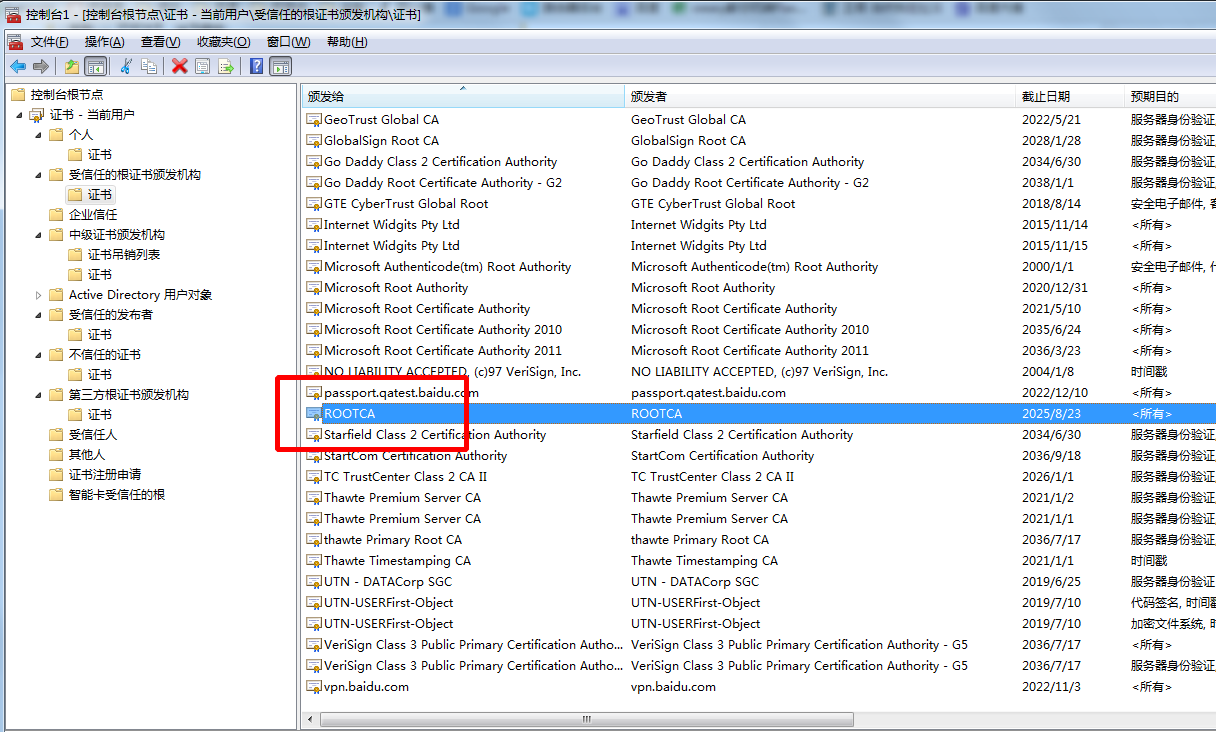
Icon

l  对于windows平台：win+R， 输入certmgr.msc

l  对于Ubuntu: 放在/usr/share/ca-certificates/中

**5.6 有风险的根证书**





没错，我发现自己的根证书列表中竟然有支付宝的alipay，这证书是啥时候跑进来的？

alipay原本没有versign的待遇的，但是自从你装了支付宝安全控件之后，他就自己插进来了，太危险了。事实上把alipay相关的根证书删掉依然是可以正常使用支付宝的，因为alipay有versign认证的证书。

再次重复一句，**整个HTTPS通信的唯一核心保障就是可信的根证书**。这种自己安装不可信的根证书会有遭到中间人攻击的风险。

**5.7 为何就不能安装看起来靠谱的根证书呢？**

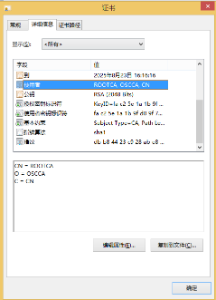
**导入ROOTCA的危险性：**

如果导入支付宝，铁道部[12306.cn](http://12306.cn/)这种第三方CA的根证书，将会大大增加受到中间人攻击的可能性。如果支付宝私钥泄露，所有支付宝用户都有遭到中间人攻击的可能性。

而铁道部的根证书更危险，铁道部的私钥必然对官方是公开的，GFW完全有能力通过铁道部的私钥来实现对所有导入了铁道部CA根证书的用户的中间人攻击，让你的HTTPS加密流量完全暴露给GFW。正确的做法是，不导入根证书，访问12306购票时候，无视证书警告，点仍然继续，依然可以正常购票.

不止如此，安装个支付宝控件，你会发现多了一个OSCCA的ROOTCA，然后你查找OSCCA，结果是：国家商用密码办公室...

如果GFW之类的代理服务器想要窃听你的https内容，就会像Fiddler一样，向所有你访问的域名颁发一个用户证书，用这个证书来和浏览器通信，从而达到窃听目的。



劝大家，运行certmgr.msc，把奇怪的证书都清理干净吧。**添加根证书**的行为就是给自己留下非常危险的后门。