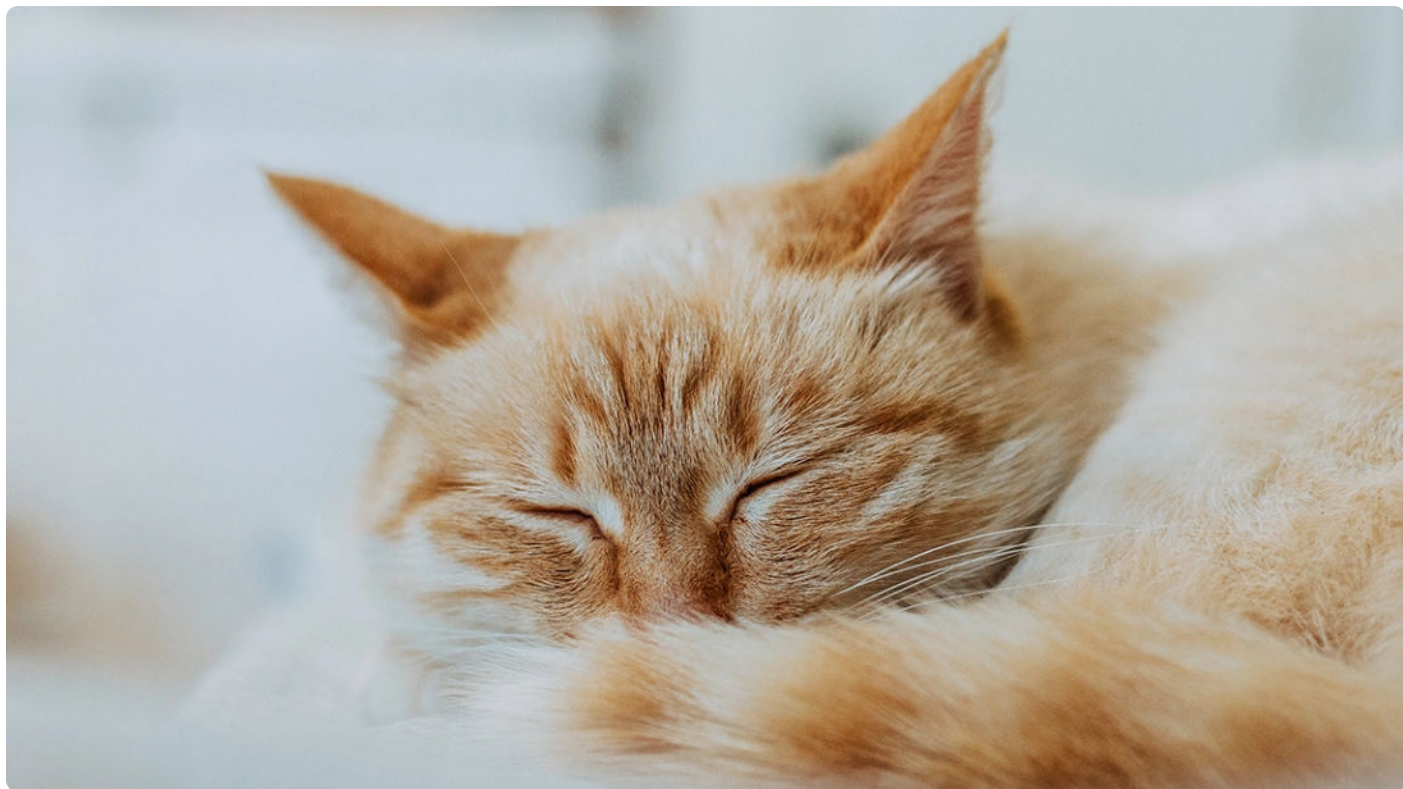


加餐六 | HTTPS：浏览器如何验证数字证书？

2019-12-20 李兵

《浏览器工作原理与实践》

课程介绍 >



讲述：李兵

时长 10:48 大小 9.91M



你好，我是李兵。

在《[🔗 36 | HTTPS：让数据传输更安全](#)》这篇文章中，我们聊了下面几个问题：

- HTTPS 使用了对称和非对称的混合加密方式，这解决了数据传输安全的问题；
- HTTPS 引入了中间机构 CA，CA 通过给服务器颁发数字证书，解决了浏览器对服务器的信任问题；
- 服务器向 CA 机构申请证书的流程；
- 浏览器验证服务器数字证书的流程。



不过由于篇幅限制，关于“**浏览器如何验证数字证书**”的这个问题我们并没有展开介绍。那么今天我们就继续聊一聊这个问题。了解了这个问题，可以方便我们把完整的 HTTPS 流程给串起来，无论对于我们理解 HTTPS 的底层技术还是理解业务都是非常有帮助的。

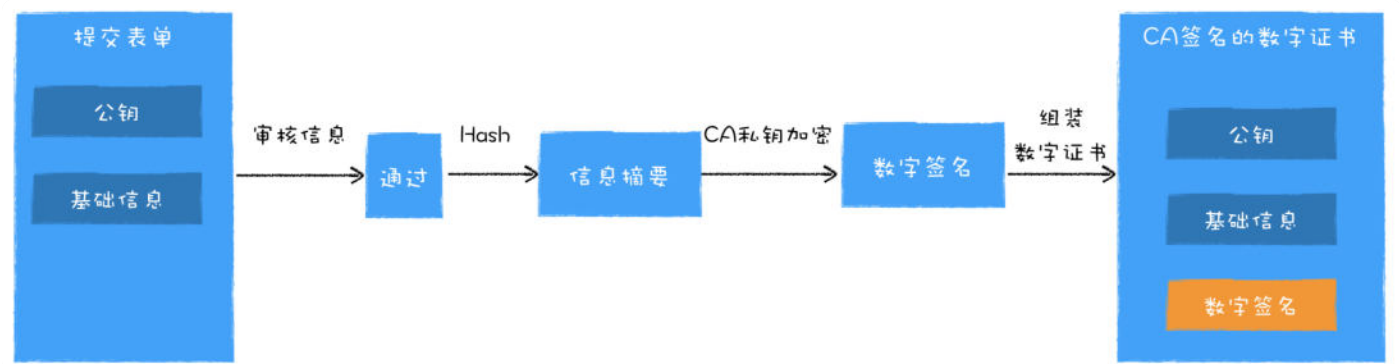
因为本文是第 36 讲的延伸，所以在分析之前，我们还是有必要回顾下**数字证书申请流程**和**浏览器验证证书的流程**，同时你最好也能回顾下第 36 讲。

数字证书申请流程

我们先来回顾下数字证书的申请流程，比如极客时间向一个 CA 机构申请数字证书，流程是什么样的呢？

首先极客时间填写了一张含有**自己身份信息**的表单，身份信息包括了自己公钥、站点资料、公司资料等信息，然后将其提交给了 CA 机构；CA 机构会审核表单中内容的真实性；审核通过后，CA 机构会拿出自己的私钥，对表单的内容进行一连串操作，包括了对明文资料进行 Hash 计算得出信息摘要，利用 CA 的私钥加密信息摘要得出数字签名，最后将数字签名也写在表单上，并将其返还给极客时间，这样就完成了一次数字证书的申请操作。

大致流程你也可以参考下图：

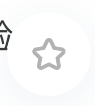


数字证书申请过程

浏览器验证证书的流程

现在极客时间的官网有了 CA 机构签发的数字证书，那么接下来就可以将数字证书应用在 HTTPS 中了。

我们知道，在浏览器和服务器建立 HTTPS 链接的过程中，浏览器首先会向服务器请求数字证书，之后浏览器要做的第一件事就是验证数字证书。那么，这里所说的“验证”，它到底是在验证什么呢？



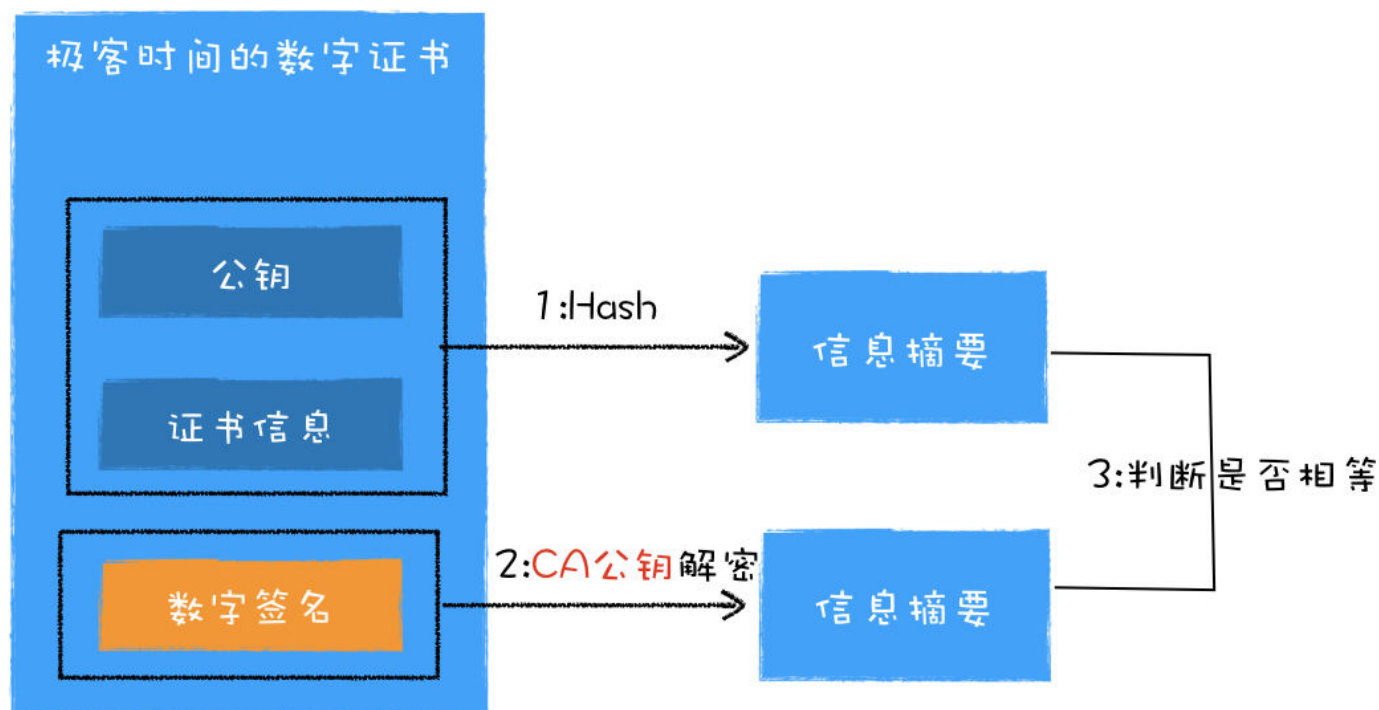
具体地讲，浏览器需要验证证书的有效期、证书是否被 CA 吊销、证书是否是合法的 CA 机构颁发的。

数字证书和身份证一样也是有时间期限的，所以**第一部分就是验证证书的有效期**，这部分比较简单，因为证书里面就含有证书的有效期，所以浏览器只需要判断当前时间是否在证书的有效期限范围内即可。

有时候有些数字证书被 CA 吊销了，吊销之后的证书是无法使用的，所以**第二部分就是验证数字证书是否被吊销了**。通常有两种方式，一种是下载吊销证书列表 -CRL (Certificate Revocation Lists)，第二种是在线验证方式 -OCSP (Online Certificate Status Protocol)，它们各有优缺点，在这里我就不展开介绍了。

最后，还要**验证极客时间的数字证书是否是 CA 机构颁发的**，验证的流程非常简单：

- 首先，浏览器利用证书的原始信息计算出信息摘要；
- 然后，利用 **CA 的公钥**来解密数字证书中的**数字签名**，解密出来的数据也是信息摘要；
- 最后，判断这两个信息摘要是否相等就可以了。



通过这种方式就验证了数字证书是否是由 CA 机构所签发的，不过这种方式又带来了一个新的疑问：**浏览器是怎么获取到 CA 公钥的？**

浏览器是怎么获取到 CA 公钥的？

通常，当你部署 HTTP 服务器的时候，除了部署当前的数字证书之外，还需要部署 CA 机构的数字证书，CA 机构的数字证书包括了 CA 的公钥，以及 CA 机构的一些基础信息。

因此，极客时间服务器就有了两个数字证书：

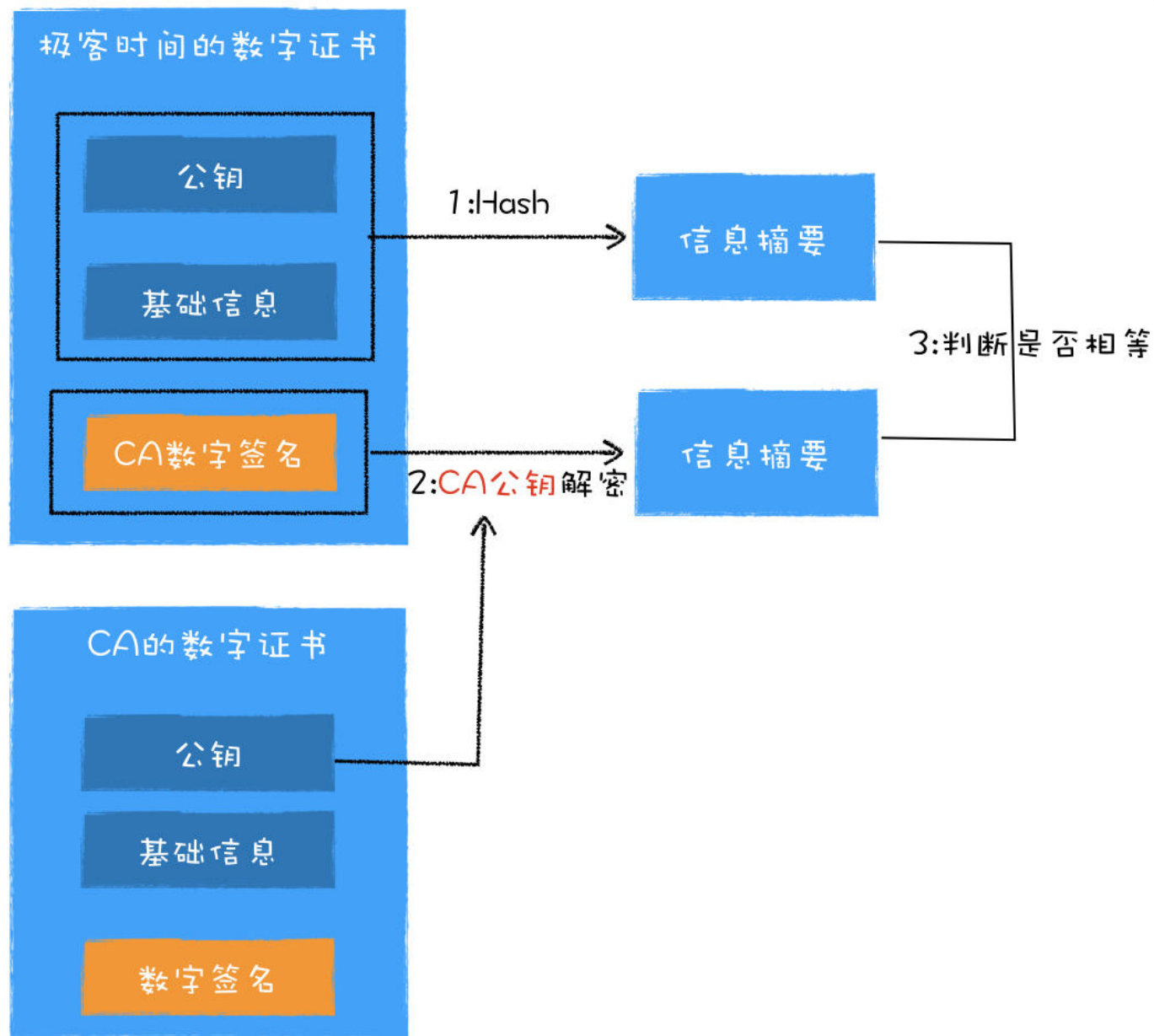
- 给极客时间域名的数字证书；
- 给极客时间签名的 CA 机构的数字证书。

然后在建立 HTTPS 链接时，服务器会将这两个证书一同发送给浏览器，于是浏览器就可以获取到 CA 的公钥了。

如果有些服务器没有部署 CA 的数字证书，那么浏览器还可以通过网络去下载 CA 证书，不过这种方式多了一次证书下载操作，会拖慢首次打开页面的请求速度，一般不推荐使用。

现在浏览器端就有了极客时间的证书和 CA 的证书，完整的验证流程就如下图所示：





CA 证书

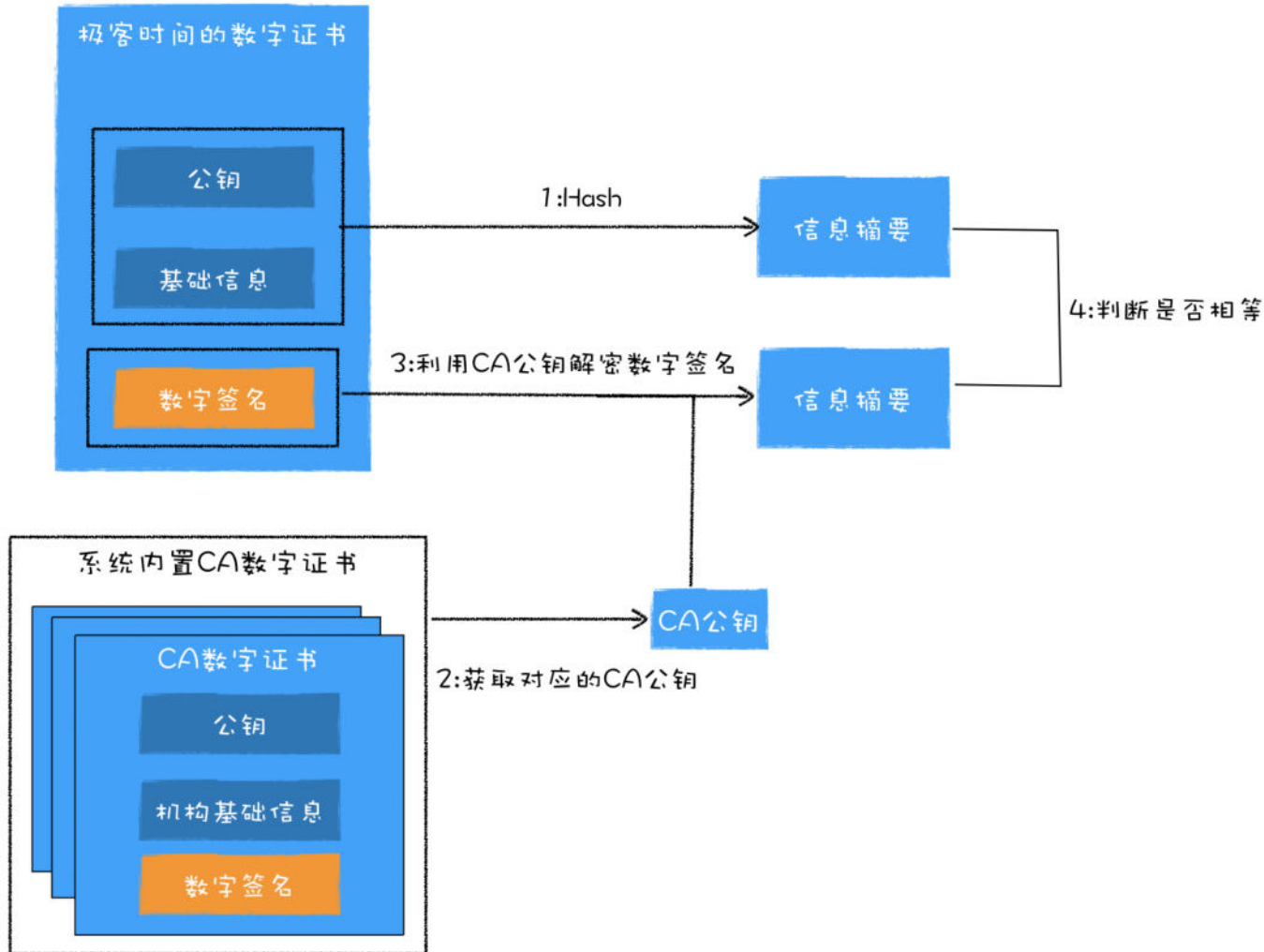
我们有了 CA 的数字证书，也就可以获取得 CA 的公钥来验证极客时间数字证书的可靠性了。

解决了获取 CA 公钥的问题，新的问题又来了，如果这个证书是一个恶意的 CA 机构颁发的怎么办？所以我们还需要浏览器证明这个 CA 机构是个合法的机构。

证明 CA 机构的合法性

这里并没有一个非常好的方法来证明 CA 的合法性，妥协的方案是，直接在操作系统中内置这些 CA 机构的数字证书，如下图所示：





操作系统内部内置 CA 数字证书

我们将所有 CA 机构的数字证书都内置在操作系统中，这样当需要使用某 CA 机构的公钥时，我们只需要依据 CA 机构名称，就能查询到对应的数字证书了，然后再从数字证书中取出公钥。

可以看到，这里有一个假设条件，浏览器默认信任操作系统内置的证书为合法证书，虽然这种方式不完美，但是却是最实用的一个。

不过这种方式依然存在问题，因为在实际情况下，CA 机构众多，因此操作系统不可能将每家 CA 的数字证书都内置进操作系统。

数字证书链


于是人们又想出来一个折中的方案，将颁发证书的机构划分为两种类型，**根 CA(Root CAs)**和**中间 CA(Intermediates CAs)**，通常申请者都是向中间 CA 去申请证书的，而根 CA 作用就





是给中间 CA 做认证，一个根 CA 会认证很多中间的 CA，而这些中间 CA 又可以去认证其他的中间 CA。


因此，每个根 CA 机构都维护了一个树状结构，一个根 CA 下面包含多个中间 CA，而中间 CA 又可以包含多个中间 CA。这样就形成了一个证书链，你可以沿着证书链从用户证书追溯到根证书。

比如你可以在 Chrome 上打开极客时间的官网，然后点击地址栏前面的那把小锁，你就可以看到 *.geekbang.org 的证书是由中间 CA GeoTrust RSA CA2018 颁发的，而中间 CA GeoTrust RSA CA2018 又是由根 CA DigiCert Global Root CA 颁发的，所以这个证书链就是：*.geekbang.org—>GeoTrust RSA CA2018—>DigiCert Global Root CA。你可以参看下图：

 DigiCert Global Root CA

↳  GeoTrust RSA CA 2018

↳  *.geekbang.org



DigiCert Global Root CA

根证书颁发机构

过期时间：2031年11月10日 星期一 台北标准时间 上午8:00:00

✔ 此证书有效

▼ 细节

主题名称

国家或地区

组织

组织单位

常用名称

颁发者名称

国家或地区

组织

组织单位

常用名称

US

DigiCert Inc

www.digicert.com

DigiCert Global Root CA


US

DigiCert Inc

www.digicert.com

DigiCert Global Root CA

好



因此浏览器验证极客时间的证书时，会先验证 *.geekbang.org 的证书，如果合法，再验证中间 CA 的证书，如果中间 CA 也是合法的，那么浏览器会继续验证这个中间 CA 的根证书。

到了这里，依然存在一个问题，那就是**浏览器怎么证明根证书是合法的？**

如何验证根证书的合法性

其实浏览器的判断策略很简单，它只是简单地判断这个根证书在不在操作系统里面，如果在，那么浏览器就认为这个根证书是合法的，如果不在，那么就是非法的。

如果某个机构想要成为根 CA，并让它的根证书内置到操作系统中，那么这个机构首先要通过 WebTrust 国际安全审计认证。

什么是 WebTrust 认证？

WebTrust 是由两大著名注册会计师协会 AICPA（美国注册会计师协会）和 CICA（加拿大注册会计师协会）共同制定的安全审计标准，主要对互联网服务商的系统及业务运作逻辑安全性、保密性等共计七项内容进行近乎严苛的审查和鉴证。只有通过 WebTrust 国际安全审计认证，根证书才能预装到主流的操作系统，并成为可信的认证机构。

目前通过 WebTrust 认证的根 CA 有 Comodo、geotrust、rapidssl、symantec、thawte、digicert 等。也就是说，这些根 CA 机构的根证书都内置在各大操作系统中，只要能从数字证书链往上追溯到这几个根证书，浏览器就会认为使用者的证书是合法的。

总结

好了，今天的内容就介绍到这里，下面我们总结下本文的主要内容：

我们先回顾了数字证书的申请流程，接着我们重点介绍了浏览器是如何验证数字证书的。

首先浏览器需要 CA 的数字证书才能验证极客时间的数字证书，接下来我们需要验证 CA 证书的合法性，最简单的方法是将 CA 证书内置在操作系统中。



不过 CA 机构非常多，内置每家的证书到操作系统中是不现实的，于是我们采用了一个折中的策略，将颁发证书的机构划分为两种类型，**根 CA(Root CAs)**和**中间 CA(Intermediates CAs)**，通常申请者都是向中间 CA 去申请证书的，而根 CA 作用就是给中间 CA 做认证，一个根 CA 会认证很多中间的 CA，而这些中间 CA 又可以去认证其他的中间 CA。

于是又引出了数字证书链，浏览器先利用中间 CA 的数字证书来验证用户证书，再利用根证书来验证中间 CA 证书的合法性，最后，浏览器会默认相信内置在系统中的根证书。不过要想在操作系统内部内置根证书却并不容易，这需要通过 WebTrust 认证，这个认证审核非常严格。

通过分析这个流程可以发现，浏览器默认信任操作系统内置的根证书，这也会带来一个问题，如果黑客入侵了你的电脑，那么黑客就有可能往你系统中添加恶意根数字证书，那么当你访问黑客站点的时候，浏览器甚至有可能会提示该站点是安全的。

因此，HTTPS 并非是绝对安全的，采用 HTTPS 只是加固了城墙的厚度，但是城墙依然有可能被突破。


课后思考

今天留给你的任务是复述下浏览器是怎么验证数字证书的，如果中间卡住了，欢迎在留言区提问交流。

感谢阅读，如果你觉得这篇文章对你有帮助的话，也欢迎把它分享给更多的朋友。

分享给需要的人，Ta订阅超级会员，你将得 50 元

Ta单独购买本课程，你将得 20 元

 生成海报并分享

 赞 7  提建议

© 版权归极客邦科技所有，未经许可不得传播售卖。页面已增加防盗追踪，如有侵权极客邦将依法追究其法律责任。



JVM + NIO + Spring

各大厂面试题及知识点详解

限时免费



精选留言 (24)

写留言



世界和平

2019-12-20

工作两年，对很多前端知识有的还是比较乱的，一知半解禁不住深究，老师的课程帮助很好的梳理了这些知识，也详细的讲解了，让我有了系统的认知，也为之后的继续学习提供了方向，非常的感谢老师，很值。已经推荐给朋友，以后如果老师再出课，也会继续跟着学习。我不是托，我不是托，我就是真诚的表示一下感谢。谢谢 ~

作者回复: 🙏



29



pacos

2020-01-05

期待老师的 Promise 加餐



9



成楠Peter

2020-01-01

这篇文章就解决了客户端验证服务器正确性的问题。但是我有一个小疑问，如果我伪造了一个客户端，同时拿到数字签名和CA公钥，通过CA公钥解密数字信息，这样是否能骗取服务端的信任？老师可以讲讲这中间的细节吗？



**林浩**

2020-10-10

总结：

浏览器怎么验证证书？

一般通过 验证证书有效期， 证书是否被CA吊销， 证书是否是合法CA机构颁发

如何验证证书有效期？

证书里面包含有效期

如何获知证书被吊销？

1. 下载吊销证书列表 2. 在线验证

如何证明是合法CA机构？

1. 通过证书原始信息（hash）计算消息摘要
2. 利用CA公钥解密证书中的数字签名，得到消息摘要
3. 将两者进行对比

浏览器怎么拿到CA公钥？

服务器部署时，除了当前数字证书外，还需要部署CA证书，CA证书上就包含了CA公钥，当建立HTTPS连接时，服务器会往浏览器发送两个证书，如果服务器上没有部署CA证书，浏览器会通过网络下载CA证书，也可以拿到CA公钥

这里只证明了CA公钥的来源，怎么知道它是合法机构？

很遗憾没有！退而求其次，计算机操作系统内置了一些颁发证书的机构，但因为机构众多不可能这么处理，所以将证书分成了“根CA”和“中间CA”，一个“根CA”会有多个“中间CA”，“根”给“中间”做认证

怎么知道根证书的合法性？

要成为“根CA”需要得到“Web Trust”认证通过才会内置到操作系统中，Web Trust 包括两个机构（AICPA【美国注册会计师协会】和 CICA【加拿大注册会计师协会】）

如果操作系统被入侵如何保证跟证书合法性？

凉凉。。。



3

**Geek_c9436e**

2020-09-15

我看完了，酣畅淋漓的感觉，满满干货，意犹未尽啊，给老师点赞，希望继续学习老师的课！



2

**雨儿**

2020-01-11

老师太好了，每天都会看看，是否老师有新的更新，期待老师不定期能更新一些

作者回复: 现在在写新专栏，暂时没办法更新了浏览器专栏了



2

**成楠Peter**

2020-01-01

我感觉老师这篇是看到了我之前的留言，专门延伸的一篇文章，点赞！

共 1 条评论 >



2

**极客时间**

2019-12-25

盗版的操作系统也有可能安装了恶意根证书啊，所以大家支持正版吧

共 1 条评论 >



3

**LEON**

2019-12-22

如果有些服务器没有部署 CA 的数字证书，那么浏览器还可以通过网络去下载 CA 证书，不过这种方式多了一次证书下载操作，会拖慢首次打开页面的请求速度，一般不推荐使用。

老师这块没听明白？这是默认因为吗？下载的是中间CA证书吗？去网络中什么地方下载？如何确保下载CA的有效性？

感谢。

共 1 条评论 >



1

**雷厉**

2021-08-07

感谢老师的分享

**undefined**

2021-03-26

看完了 感谢分享





陈启航

2021-02-23

值得之后前端开发经验更多之后 回来重读



灵感_idea

2021-01-23

学完打个卡，老师讲的挺全面的，虽然很多地方稍显粗略，但或许是更利于接受的，还是要多学几遍，反复琢磨。



子曰

2020-11-16

通过图解的方式把一些底层的原理阐述的很清晰，老师辛苦，干货满满的课程👍👍👍👍👍👍



5102

2020-08-25

虽然之前有到处查询百度过这些知识，到目前为止，这个专栏让我彻底重新认识浏览器等，也理清很多的疑惑，真是醍醐灌顶的感觉，非常赞，v8那篇我还是会继续订阅的。



后脑勺

2020-05-30

完结，撒贝宁

共 1 条评论 >



tt

2020-05-21

中间CA众多，如果证书链很长的话，浏览器对每一个中间证书都需要发起https请求去中间网站的CA获取么？



Learning

2020-03-28

谢谢老师的加餐



Lorin

2020-02-23



老师开一个前端专栏吧，前端领域里面找一个如此高质量的课程简直是太少了。



淡

2020-01-09

很好的解答了客户端是如何拿到CA公钥以及根CA的存储问题。说好的promise呢，哈哈。
题外话，极客时间课程更新没提示了，之前都有的。不知道是不是因为课程标记为”选学“的原因。

