**数字证书**

**常见的抵赖行为**

1. A向B发了信息M，事后，A不承认其曾经发过。
2. A向B发了信息M0，事后，A却说其发了信息M1。
3. B收到了A发来的信息M，事后，B不承认收到了。
4. B收到了A发来的信息M0，事后，B却说收到的是M1。

一般的公钥是不会采用明文格式传输给别人，在正常情况下都使用一个文件，这个文件就是公钥文件，然后这个文件可以交给其他人用于加密，但是如果在传输过程中有人恶意破坏，将你的公钥偷换成了他的公钥，最终使用公钥的一方使用了做恶者的公钥进行数据加密，这样做恶者就可以用他自己的密钥解密看到数据了。

为了解决这个问题，需要一个公证方来颁发证书，这样任何人都可以验证公钥是谁发的。这就是CA，CA确认公钥的原理也很简单，它将它自己的公钥发布给所有人，然后一个想要发布自己公钥的人可以将自己的公钥和一些身份信息发给CA，CA用自己的私钥进行加密，这里称为签名。然后这个包含了你的公钥和你的信息的文件就称为证书文件了。

这样所有得到这些证书文件的人，通过CA发布的公钥来解密文件，如果能正常解密那么证书里面的信息就一定是真的，因为加密方只可能是CA，其他人没有CA的密钥。这样当你解开公钥文件，看看里面的信息就知道这是你需要使用加密的公钥了。

|  |
| --- |
| 命令openssl version -a确定OPENSSLDIR目录，通常在C:\OpenSSL-Win32\bin。  查看openssl.cfg文件，CA\_default部分中的dir定义了CA的管理目录。  mkdir demoCA目录，并在demoCA目录下增加certs，crl，newcerts，private目录，增加serial，index.txt文件，echo 01>>serial。 |
| 以下命令需要在C:\OpenSSL-Win32\bin目录下执行。 |

**CA自签署**

|  |
| --- |
| **生成私钥cakey.pem**  openssl genrsa -out demoCA\private\cakey.pem 2048 |
| **生成CA根证书cacert.pem**  openssl req -new -x509 -key demoCA\private\cakey.pem -out demoCA\cacert.pem -days 365  Country Name默认CN  State or Province Name填写BJ  Locality Name空白不填  Organization Name填写TH  Organizational Unit Name空白不填  Common Name填写www.bj.com  Email Address空白不填 |

**颁发证书**

|  |
| --- |
| 1，首先生成私钥openssl genrsa -out demoCA\ssltlskey.pem 2048 |
| 2，生成签署请求openssl req -new -key demoCA\ssltlskey.pem -out demoCA\ssltlscert.csr -days 365  Country Name默认CN  State or Province Name填写GD  Locality Name空白不填  Organization Name填写ZS  Organizational Unit Name空白不填  Common Name填写192.168.80.89  Email Address空白不填  A challenge password空白不填  An optional company name空白不填 |
| 3，生成ssltlscert.crt证书  openssl ca -in demoCA\ssltlscert.csr -out demoCA\ssltlscert.crt -days 365  Sign the certificate?填写y  1 out of 1 certificate requests certified, commit? 填写y  当出现错误提示needed to be the same时，可以更改openssl.cfg中的match为optional  [ policy\_match ]  countryName = match  stateOrProvinceName = optional  organizationName = optional  organizationalUnitName = optional  commonName = supplied  emailAddress = optional |
| 4，查看ssltlscert.crt证书  openssl x509 -in demoCA\ssltlscert.crt -noout -serial -subject |

**鉴定证书**

|  |
| --- |
| 第1种方法  openssl verify –Cafile demoCA\cacert.pem demoCA\ssltlscert.crt |
| 第2种方法  mkdir CApath  openssl x509 -hash -in demoCA\cacert.pem -noout  假设得到hash值为24b24fae  copy demoCA\cacert.pem CApath\24b24fae.0  openssl verify –CApth CApath demoCA\ssltlscert.crt |

**证书格式**

|  |
| --- |
| **格式转换**   1. 转换为ssltlscert.pfx格式。pfx格式的数字证书是包含有私钥的，所以需要输入密码对私钥再次加密。   openssl pkcs12 -export -out demoCA\ssltlscert.pfx -inkey demoCA\ssltlskey.pem -in demoCA\ssltlscert.crt  Enter Export Password: 空白不填  Verifying - Enter Export Password: 空白不填   1. 转换为ssltlscert.cer格式。cer格式的数字证书只有公钥没有私钥。   openssl x509 -inform pem -in demoCA\ssltlscert.crt -outform der -out demoCA\ssltlscert.cer  查看证书信息  openssl x509 -in demoCA\ssltlscert.cer -text -noout  若报错unable to load certificate则说明打开的是der格式，执行以下命令查看  openssl x509 -in demoCA\ssltlscert.cer -inform der -text -noout |
| **常用格式：**  .DER .CER，文件是二进制格式，只保存证书，不保存私钥。  .PEM，一般是文本格式，可保存证书，可保存私钥。  .CRT，可以是二进制格式，可以是文本格式，与 .DER 格式相同，不保存私钥。  .PFX .P12，二进制格式，同时包含证书和私钥，一般有密码保护。  .JKS，二进制格式，同时包含证书和私钥，一般有密码保护。 |

**撸免费证书**

|  |
| --- |
| https://github.com/acmesh-official/acme.sh |
| https://md.lan.ute.pub/ |
| certbot,caddy,traefik,cert manager |
|  |

**关闭windows根证书更新**

|  |
| --- |
|  |

**wireshark解密SSL**

|  |
| --- |
| 方法1，导入服务器证书  打开Edit/Preferences/Protocols/SSL，然后点击RSA keys list:Edit，创建一个新的RSAkey，  其中IP address是服务器的IP，prot一般是443，protocol一般是http，keyfile选择服务器的rsa key文件，这个必须是解密的pem格式。  此方法只能解密通过RSA交换的密钥。 |
| 方法2，截取master\_secret  环境变量中，新建SSLKEYLOGFILE=D:\ssl.log  然后打开Edit/Preferences/Protocols/SSL，在master-secret log filename可以看到该文件。 |