How X.509 Works

2014-10-26

YiWang ZHENG(12330423)

In cryptography, X.509 is an ITU-T standard for a public key infrastructure and Privilege Management Infrastructure. X.509 specifies, amongst other things, standard formats for public key certificates, certificate revocation lists, attribute certificates, and a certification path validation algorithm. X.509 is usually referred to when discussing about CA certification In this document, I will give an example of X.509 certificate and analyze it in detail.

- Why use certificate

  If we don't perform any security protection between server and client, all the message will be transported in plaintext, that can be very dangerous, and the problem is we can not easily just encrypt the message with some encryption algorithm. For example:

  1. Alice send a request she want to login
  2. server give a public key to Alice
  3. but the public key was hold up by someone
  4. this guy pretend to be official server and give a new public key to Alice
  5. Alice get the fake public key and encrypt her username and password of this webbank and send out.
  6. The guy hold up the request again and use his private key to decrypt the message and get Alice username and password.

  If we use certificate we can prevent this kind of attack as certificate and help us to authenticate the communicating parties.

- Generate a X.509 certificate

  generate a private key of 1024 bit and write it to file tmp:

```
zheng@Insprion:~/文档/websecurity/X509$ openssl genrsa 1024 > tmp
Generating RSA private key, 1024 bit long modulus
.....................................................++++++
............++++++
e is 65537 (0x10001)
```

genera a certificate with the key saved in tmp and write to file certificate-x509:

```
zheng@Insprion:~/文档/websecurity/X509$ openssl req -x509 -new -key tmp > certificate-x5
09
You are about to be asked to enter information that will be incorporated
into your certificate request.
What you are about to enter is what is called a Distinguished Name or a DN.
There are quite a few fields but you can leave some blank
For some fields there will be a default value,
If you enter '.', the field will be left blank.
-----
Country Name (2 letter code) [AU]:CN
State or Province Name (full name) [Some-State]:Guangdong
Locality Name (eg, city) []:Guangzhou
Organization Name (eg, company) [Internet Widgits Pty Ltd]:School of Software, SYSU
Organizational Unit Name (eg, section) []:Class 2, Computer Science and Application
Common Name (e.g. server FQDN or YOUR name) []:Yiwang ZHENG
Email Address []:940922457@qq.com
zheng@Insprion:~/文档/websecurity/X509$
```

print the certificate in a pretty way and write to file certificate-x509-pretty:

```
zheng@Insprion:~/文档/websecurity/X509$ openssl x509 -in certificate-x509 -text > certif
icate-x509-pretty
zheng@Insprion:~/文档/websecurity/X509$
```

- Certificate structure

From the content in file certificate-x509-pretty we can know that A X.509 digital certificate has the following structure

- Certificate
  - Version
  - Serial Number
  - Algorithm ID
  - Issuer
  - Validity
    - Not Before
    - Not After
  - Subject
  - Subject Public Key Info
    - Public Key Algorithm
    - Subject Public Key
  - Issuer Unique Identifier (optional)
  - Subject Unique Identifier (optional)
  - Extensions (optional)
    - ...
- Certificate Signature Algorithm
- Certificate Signature

For some important key, the corresponding value of the certificate generated at the beginning are list in the follow table:

Key	Value	Meaning
Version	3(0x2)	Version 3 of X.509
Serial Number	14321780464519547736 (0xc6c12f73d2d10b58)	A unique number to identify a X.509 certificate
Issuer	C=CN, ST=Guangdong, L=Guangzhou, O=School of Software, SYSU, OU=Class 2, Computer Science and Application, CN=Yiwang ZHENG/emailAddress=940922457@qq.com	Some information of the user who give out this certificate
Validaty	Not Before: Oct 26 03:27:18 2014 GMT Not After : Nov 25 03:27:18 2014 GMT	The validate date of thi certificate
Subject	The same as Issuer here	Information about the user of this certificate
Subject Public Key Info	Public Key Algorithm: rsaEncryption Public-Key: (1024 bit) Modulus: 00:a4:8e:97:58:49:65:f7:49:8f:e8:f9:ff:0e:f8: 38:e7:6b:a3:2a:82:2c:b5:26:e5:f6:15:b1:a2:7d: 97:63:9c:81:bf:f8:41:46:f9:28:8c:be:bb:8f:a4: 16:43:58:95:04:92:15:fd:5e:dc:65:5a:b8:ff:2b: 90:11:d2:a9:79:40:f6:03:df:1b:c4:d7:5c:3a:47: 97:ff:b3:3c:63:2e:f7:66:b3:4c:65:65:a8:2b:94: 79:b3:05:cf:69:34:af:99:5d:19:d8:23:b4:5c:c0: de:2e:33:0d:a0:c8:7c:dd:73:4c:67:8f:6b:ab:8d: 04:e7:95:37:94:14:6a:57:d3	The information about the user public key, including the encryption algorithm, the length, and the key content.
X509v3 extensions	X509v3 Subject Key Identifier: 17:B8:4F:BC:67:62:83:25:46:53:56:5B:F9:09:A8:76:9A:27:99:C4 X509v3 Authority Key Identifier: keyid:17:B8:4F:BC:67:62:83:25:46:53:56:5B:F9:09:A8:76:9A:27:99:C4 X509v3 Basic Constraints: CA:TRUE	The key id for subject key and authority key
Signature Algorithm	Sha256WithRSAEncryption 6b:c3:a7:39:e9:1b:95:f5:96:84:27:db:ad:71:50:b8:1f:10: d9:2b:50:3a:36:e4:7f:fb:32:a0:12:c0:a2:ca:11:8b:3a:f7: c6:be:e7:43:f4:7a:1c:45:f1:4e:34:c7:c4:ff:08:44:f8:27: 36:ea:78:3d:03:13:97:af:18:25:2a:8b:12:2d:72:9d:dd:91: a5:d7:e5:cc:dc:82:48:e6:25:bf:82:3f:1d:e3:a6:99:82:8e: 12:98:08:77:93:ca:17:68:78:f7:e2:f9:d3:35:0a:07:bb:9c: ef:08:e2:94:5f:3f:11:e8:d4:f8:5e:0a:e1:a6:7c:70:ef:d2: 03:b1	State the encryption algorithm of the signature , the following is the content of the signature

- How to perform authenticate(Bob send message to Alice and Alice want to make sure she is communicating with Bob)
    1. Alice received the message.
    2. Alice use her own private key to decrypt the message.
    3. Use Bob's public key to get the digest of the original message from the digital signature.
    4. Alice perform a hash function on the data after decrypted and get a digest for the received data.
    5. Alice compare two digest, if they are the same, then she can make sure she is communicating with Bob.

  By doing this operation, the process of certification can satisfy these property:

    1. Security – only the receiver can read the message.
    2. Certification – receiver can confirm the sender's identity.
    3. Integrity – receiver can check whether the message he receive is the original message the sender sent, if the message has been tampered by someone else during delivery, it can't pass the authentication.
    4. Non-Repudiation – the sender can not deny that he has sent some message out.