

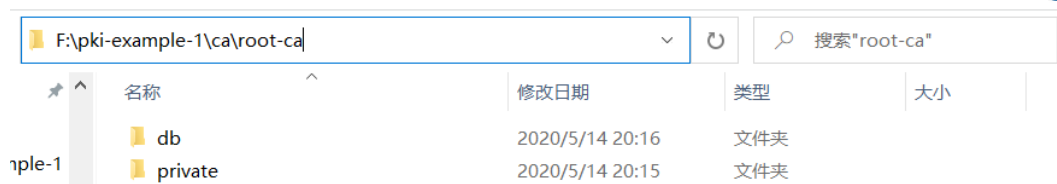
前期准备和一些废话：

- 1.环境：windows10，把 openssl 配置好；linux 直接参考 OpenSSL PKI Tutorial，
- 2.获取 Simple PKI 示例文件
方式 1：通过 git 直接获取，关于 git 的相关操作略
git clone <https://bitbucket.org/stefanholek/pki-example-1>
方式 2：直接打开上面网址获得，具体操作略
- 3.新建一个文件夹用于存放文件，将刚才的获得的 etc 文件夹放入，我直接用 git 下来的文件夹 pki-example-1 就没有这一步
- 4.下文中的命令行所涉及的路径和文件名请自行根据需求更改，命令行这东西踩的坑多了就会用了
5. 下文中 1.1、1.2、2.1、2.2 必须在使用 openssl ca 命令之前完成，其他创建文件夹步骤根据个人习惯进行自行调整
6. 本着前人踩坑后人白嫖的原则写这篇教程，也是培养一下自己写博客的习惯，由于是做完后回忆着写，一些不是重点的步骤难免有些混乱

0.cmd 进入刚才创建的存放文件目录，输入 openssl 进入 openssl，或者下面的命令前加上 openssl

1.创建 rootCA

1.1 进入刚才的存放文件目录，开始手动创建目录，最后所需要的文件夹路径如：



1.2 db 文件夹中创建图中红框文件

root-ca.crl.srl	2020/5/14 9:21	SRL 文件	1 KB
root-ca.crt.srl	2020/5/14 20:16	SRL 文件	1 KB
root-ca.crt.srl.old	2020/5/14 20:16	OLD 文件	1 KB
root-ca.db	2020/5/14 20:16	Data Base File	1 KB
root-ca.db.attr	2020/5/14 20:16	ATTR 文件	1 KB
root-ca.db.attr.old	2020/5/14 20:16	OLD 文件	1 KB
root-ca.db.old	2020/5/14 20:16	OLD 文件	1 KB

root-ca.crt.srl、root-ca.crl.srl 用记事本或者 notepad++打开输入 01，用于表示证书序号,CRL 号码从 01 开始

1.3 按实验要求更改配置文件 root-ca.conf

```
指令.txt x root-ca.conf x
1 # Simple Root CA
2
3 # The [default] section contains global constants that can be referred to from
4 # the entire configuration file. It may also hold settings pertaining to more
5 # than one openssl command.
6
7 [ default ]
8 ca = root-ca # CA name
9 dir = . # Top dir
10
11 # The next part of the configuration file is used by the openssl req command.
12 # It defines the CA's key pair, its DN, and the desired extensions for the CA
13 # certificate.
14
15 [ req ]
16 default_bits = 2048 # RSA key size
17 encrypt_key = yes # Protect private key
18 default_md = sha1 # MD to use
19 utf8 = yes # Input is UTF-8
20 string_mask = utf8only # Emit UTF-8 strings
21 prompt = no # Don't prompt for DN
22 distinguished_name = ca_dn # DN section
23 req_extensions = ca_reqext # Desired extensions
24
25 [ ca_dn ]
26 0.domainComponent = "edu"
27 1.domainComponent = "gxun"
28 organizationName = "软件学院"
29 organizationalUnitName = "18信安"
30 commonName = "rootCA"
31
32 [ ca_reqext ]
33 keyUsage = critical,keyCertSign,cRLSign
34 basicConstraints = critical,CA:true
35 subjectKeyIdentifier = hash
```

1.4 创建 rootCA 请求, cmd 输入

```
req -new \
    -config etc/root-ca.conf \
    -out ca/root-ca.csr \
    -keyout ca/root-ca/private/root-ca.key
```

1.4 生成 rootCA 证书, cmd 输入

```
ca -selfsign \
    -config etc/root-ca.conf \
    -in ca/root-ca.csr \
    -out ca/root-ca.crt \
    -extensions root_ca_ext
```

```

ca 管理员: 命令提示符 - openssl

F:\pki-example-1>openssl
OpenSSL> req -new \
> -config etc/root-ca.conf \
> -out ca/root-ca.csr \
> -keyout ca/root-ca/private/root-ca.key
Generating a RSA private key
.....+++++
.....+++++
writing new private key to 'ca/root-ca/private/root-ca.key'
Enter PEM pass phrase:
Verifying - Enter PEM pass phrase:
-----
OpenSSL> ca -selfsign \
> -config etc/root-ca.conf \
> -in ca/root-ca.csr \
> -out ca/root-ca.crt \
> -extensions root_ca_ext
Using configuration from etc/root-ca.conf
Enter pass phrase for ./ca/root-ca/private/root-ca.key:
Check that the request matches the signature
Signature ok
Certificate Details:
  Serial Number: 1 (0x1)
  Validity
    Not Before: May 14 12:15:55 2020 GMT
    Not After : May 14 12:15:55 2030 GMT
  Subject:
    domainComponent           = edu
    domainComponent           = gxun
    organizationName          = \U8F6F\U4EF6\U5B66\U9662
    organizationalUnitName    = 18\U4FE1\U5B89
    commonName                 = rootCA
  X509v3 extensions:
    X509v3 Key Usage: critical
      Certificate Sign, CRL Sign
    X509v3 Basic Constraints: critical
      CA:TRUE
    X509v3 Subject Key Identifier:
      C8:18:EE:E6:D1:14:13:33:7E:7A:E2:C9:11:8B:B2:98:C4:20:DC:29
    X509v3 Authority Key Identifier:
      keyid:C8:18:EE:E6:D1:14:13:33:7E:7A:E2:C9:11:8B:B2:98:C4:20:DC:29

Certificate is to be certified until May 14 12:15:55 2030 GMT (3652 days)
Sign the certificate? [y/n]:y

1 out of 1 certificate requests certified, commit? [y/n]y
Write out database with 1 new entries
Data Base Updated

```

结果如下图，rootCA 的 root-ca.key 存放在 ca/root-ca/private 中不展示出来：

root-ca	2020/5/14 20:16	文件夹	
signing-ca	2020/5/14 20:25	文件夹	
root-ca.crt	2020/5/14 20:16	安全证书	5 KB
root-ca.csr	2020/5/14 20:15	CSR 文件	2 KB
signing-ca.crt	2020/5/14 20:16	安全证书	5 KB
signing-ca.csr	2020/5/14 20:16	CSR 文件	2 KB

2.创建 signingCA

- 2.1 创建目录，类似 1.1 操作，不赘述
- 2.2 创建数据库，类似 1.2 操作，不赘述
- 2.3 修改配置文件 signing-ca.conf，修改部分如图：

```
指令.txt signing-ca.conf
7 [ default ]
8 ca = signing-ca # CA name
9 dir = . # Top dir
10
11 # The next part of the configuration file is used by the openssl req command.
12 # It defines the CA's key pair, its DN, and the desired extensions for the CA
13 # certificate.
14
15 [ req ]
16 default_bits = 2048 # RSA key size
17 encrypt_key = yes # Protect private key
18 default_md = sha1 # MD to use
19 utf8 = yes # Input is UTF-8
20 string_mask = utf8only # Emit UTF-8 strings
21 prompt = no # Don't prompt for DN
22 distinguished_name = ca_dn # DN section
23 req_extensions = ca_reqext # Desired extensions
24
25 [ ca_dn ]
26 0.domainComponent = "edu"
27 1.domainComponent = "gxun"
28 organizationName = "软件学院"
29 organizationalUnitName = "18信安"
30 commonName = "signingCA"
31
32 [ ca_reqext ]
```

注意修改和添加下面部分，涉及到实验手册 4、5、6 步骤（图序号有错不想改）：

```
# Certificate extensions define what types of certificates the CA is able to
# create.

[ email_ext ] 3.签署电子邮箱用的证书
keyUsage = critical,digitalSignature,keyEncipherment
basicConstraints = CA:false
extendedKeyUsage = emailProtection,clientAuth
subjectKeyIdentifier = hash
authorityKeyIdentifier = keyid:always

[ identity_ext ] 4. 签署用于身份认证的证书
keyUsage = critical,digitalSignature,keyEncipherment
basicConstraints = CA:false
extendedKeyUsage = serverAuth,clientAuth
subjectKeyIdentifier = hash
authorityKeyIdentifier = keyid:always

[ encryption_ext ] 5.签署用于数据加密的证书
keyUsage = critical,digitalSignature,dataEncipherment,keyEncipherment
basicConstraints = CA:false
extendedKeyUsage = OCSPSigning,serverAuth,clientAuth,emailProtection,codeSigning,timeStamping
subjectKeyIdentifier = hash
authorityKeyIdentifier = keyid:always
```

如果没有这部分可以省略 -extensions 命令，删掉后要把 [signing-ca] 中 x509_extensions 也删了； extendKeyUsage 部分根据要求的用途进行修改，对应的证书配置文件部分一起修改

2.4 创建 signingCA 请求

```
req -new \
    -config etc/signing-ca.conf \
    -out ca/signing-ca.csr \
    -keyout ca/signing-ca/private/signing-ca.key
```

2.5 生成 signingCA 证书

```
ca \
    -config etc/root-ca.conf \
    -in ca/signing-ca.csr \
    -out ca/signing-ca.crt \
    -extensions signing_ca_ext
```

```
Data Base Updated
OpenSSL> req -new \
> -config etc/signing-ca.conf \
> -out ca/signing-ca.csr \
> -keyout ca/signing-ca/private/signing-ca.key
Generating a RSA private key
.....+++++
+++++
writing new private key to 'ca/signing-ca/private/signing-ca.key'
Enter PEM pass phrase:
Verifying - Enter PEM pass phrase:

OpenSSL> ca \
> -config etc/root-ca.conf \
> -in ca/signing-ca.csr \
> -out ca/signing-ca.crt \
> -extensions signing_ca_ext
Using configuration from etc/root-ca.conf
Enter pass phrase for ./ca/root-ca/private/root-ca.key:
Check that the request matches the signature
Signature ok
Certificate Details:
  Serial Number: 2 (0x2)
  Validity
    Not Before: May 14 12:16:52 2020 GMT
    Not After : May 14 12:16:52 2030 GMT
  Subject:
    domainComponent           = edu
    domainComponent           = gxun
    organizationName          = \U8F6F\U4EF6\U5B66\U9662
    organizationalUnitName    = 18\U4FE1\U5B89
    commonName                 = signingCA
  X509v3 extensions:
    X509v3 Key Usage: critical
      Certificate Sign, CRL Sign
    X509v3 Basic Constraints: critical
      CA:TRUE, pathlen:0
    X509v3 Subject Key Identifier:
      CB:7D:5E:45:CE:21:2C:37:BE:19:A7:DB:BD:B4:10:DE:F6:C5:BF:9A
    X509v3 Authority Key Identifier:
      keyid:C8:18:EE:E6:D1:14:13:33:7E:7A:E2:C9:11:8B:B2:98:C4:20:DC:29

Certificate is to be certified until May 14 12:16:52 2030 GMT (3652 days)
Sign the certificate? [y/n]:y

1 out of 1 certificate requests certified, commit? [y/n]y
Write out database with 1 new entries
Data Base Updated
```







结果类似 rootCA:

脑 > 本地磁盘 (F:) > pki-example-1 > ca >

↺

↻

🔍 搜索"ca"

名称	修改日期	类型	大小
 root-ca	2020/5/14 20:16	文件夹	
 signing-ca	2020/5/14 20:25	文件夹	
 root-ca.crt	2020/5/14 20:16	安全证书	5 KB
 root-ca.csr	2020/5/14 20:15	CSR 文件	2 KB
 signing-ca.crt	2020/5/14 20:16	安全证书	5 KB
 signing-ca.csr	2020/5/14 20:16	CSR 文件	2 KB

3.创建 certs 文件夹存放下面用创建好的 CA 证书来签发的证书，crl 存放吊销证书列表

ca	2020/5/14 20:16	文件夹
certs	2020/5/14 20:25	文件夹
crl	2020/5/14 20:31	文件夹
etc	2020/5/14 17:08	文件夹

4. 签署电子邮箱用的证书

4.1 修改配置文件 email.conf

```
# Email certificate request

# This file is used by the openssl req command. Since we cannot know the DN in
# advance the user is prompted for DN information.

[ req ]
default_bits          = 2048                # RSA key size
encrypt_key           = yes                  # Protect private key
default_md             = sha1                # MD to use
utf8                   = yes                 # Input is UTF-8
string_mask            = utf8only            # Emit UTF-8 strings
prompt                = no                   # Don't Prompt for DN
distinguished_name     = email_dn           # DN template
req_extensions         = email_reqext       # Desired extensions

[ email_dn ]
0.domainComponent     = "edu"
1.domainComponent     = "gxun"
organizationName      = "软件学院"
organizationalUnitName = "18信安"
commonName             = " " 这里写你的名字

[ email_reqext ]
keyUsage               = critical,digitalSignature,keyEncipherment
basicConstraints       = CA:false
extendedKeyUsage       = emailProtection,clientAuth
subjectKeyIdentifier   = hash
```

prompt=yes就可以手动
输入DN，想偷懒直接在
配置文件中写好

4.2 创建签署电子邮箱的证书请求

```
req -new \
    -config etc/email.conf \
    -out certs/email.csr \
    -keyout certs/email.key
```

4.3 生成签署电子邮箱的证书，使用 signingCA 颁发

```
ca \
    -config etc/signing-ca.conf \
    -in certs/email.csr \
    -out certs/email.crt \
    -extensions email_ext
```

```

管理员: 命令提示符 - openssl
OpenSSL> req -new \
> -config etc/email.conf \
> -out certs/email.csr \
> -keyout certs/email.key
Generating a RSA private key
.....+++++
.....+++++
writing new private key to 'certs/email.key'
Enter PEM pass phrase:
Verifying - Enter PEM pass phrase:
-----
OpenSSL> ca \
> -config etc/signing-ca.conf \
> -in certs/email.csr \
> -out certs/email.crt \
> -extensions email_ext
Using configuration from etc/signing-ca.conf
Enter pass phrase for ./ca/signing-ca/private/signing-ca.key:
Check that the request matches the signature
Signature ok
Certificate Details:
  Serial Number: 1 (0x1)
  Validity
    Not Before: May 14 12:17:59 2020 GMT
    Not After : May 14 12:17:59 2022 GMT
  Subject:
    domainComponent           = edu
    domainComponent           = gxun
    organizationName          = \U8F6F\U4EF6\U5B66\U9662
    organizationalUnitName    = 18\U4FE1\U5B89
    commonName                 = \U672F\U5B66\U9662
  X509v3 extensions:
    X509v3 Key Usage: critical
      Digital Signature, Key Encipherment
    X509v3 Basic Constraints:
      CA:FALSE
    X509v3 Extended Key Usage:
      E-mail Protection, TLS Web Client Authentication
    X509v3 Subject Key Identifier:
      4E:7E:6B:09:8C:23:DE:DB:39:EF:DE:0B:A7:0E:24:FD:79:9C:57:12
    X509v3 Authority Key Identifier:
      keyid:CB:7D:5E:45:CE:21:2C:37:BE:19:A7:DB:BD:B4:10:DE:F6:C5:BF:9A

Certificate is to be certified until May 14 12:17:59 2022 GMT (730 days)
Sign the certificate? [y/n]:y

1 out of 1 certificate requests certified, commit? [y/n]:y
Write out database with 1 new entries
Data Base Updated

```

5. 签署用于身份认证的证书

5.1 从 <https://bitbucket.org/stefanholek/pki-example-3/src/master/etc/identity.conf> 获取配置文件 identity.conf 并进行如下修改:

```

# Identity certificate request

[ req ]
default_bits          = 2048                # RSA key size
encrypt_key           = yes                  # Protect private key
default_md             = sha1                 # MD to use
utf8                   = yes                  # Input is UTF-8
string_mask           = utf8only              # Emit UTF-8 strings
prompt                = no                    # Don't Prompt for DN
distinguished_name     = identity_dn          # DN template
req_extensions         = identity_reqext      # Desired extensions

[ identity_dn ]
0.domainComponent     = "edu"
1.domainComponent     = "gxun"
organizationName      = "软件学院"
organizationalUnitName = "18信安"
commonName            = " "

[ identity_reqext ]
keyUsage              = critical,digitalSignature,keyEncipherment
basicConstraints       = CA:false
extendedKeyUsage       = serverAuth,clientAuth
subjectKeyIdentifier   = hash

```

5.2 创建用于身份认证的证书请求

```
req -new \  
-config etc/identity.conf \  
-out certs/identity.csr \  
-keyout certs/identity.key
```

5.3 生成用于身份认证的证书

```
ca \  
-config etc/signing-ca.conf \  
-in certs/identity.csr \  
-out certs/identity.crt \  
-extensions identity_ext
```

```
管理员: 命令提示符 - openssl  
OpenSSL> req -new \  
> -config etc/identity.conf \  
> -out certs/identity.csr \  
> -keyout certs/identity.key  
Generating a RSA private key  
.....+++++  
.....+++++  
writing new private key to 'certs/identity.key'  
Enter PEM pass phrase:  
Verifying - Enter PEM pass phrase:  
-----  
OpenSSL> ca \  
> -config etc/signing-ca.conf \  
> -in certs/identity.csr \  
> -out certs/identity.crt \  
> -extensions identity_ext  
Using configuration from etc/signing-ca.conf  
Enter pass phrase for ./ca/signing-ca/private/signing-ca.key:  
Check that the request matches the signature  
Signature ok  
Certificate Details:  
  Serial Number: 2 (0x2)  
  Validity  
    Not Before: May 14 12:19:08 2020 GMT  
    Not After : May 14 12:19:08 2022 GMT  
  Subject:  
    domainComponent           = edu  
    domainComponent           = gxun  
    organizationName          = \U8F6F\U4EF6\U5B66\U9662  
    organizationalUnitName    = 18\U4FE1\U5B89  
    commonName                 = \U6 7\U5 10\U6  
  X509v3 extensions:  
    X509v3 Key Usage: critical  
      Digital Signature, Key Encipherment  
    X509v3 Basic Constraints:  
      CA:FALSE  
    X509v3 Extended Key Usage:  
      TLS Web Server Authentication, TLS Web Client Authentication  
    X509v3 Subject Key Identifier:  
      43:AD:1D:01:A7:FC:80:FA:A9:C7:6D:A4:F9:99:EC:1F:80:B1:D9:BF  
    X509v3 Authority Key Identifier:  
      keyid:CB:7D:5E:45:CE:21:2C:37:BE:19:A7:DB:BD:B4:10:DE:F6:C5:BF:9A  
  
Certificate is to be certified until May 14 12:19:08 2022 GMT (730 days)  
Sign the certificate? [y/n]:y  
  
1 out of 1 certificate requests certified, commit? [y/n]:y  
Write out database with 1 new entries  
Data Base Updated
```

6. 签署用于数据加密的证书

6.1 从 <https://bitbucket.org/stefanholek/pki-example-3/src/master/etc/encryption.conf> 获取配置文件 encryption.conf 并进行如下修改:


```
# Encryption certificate request

[ req ]
default_bits          = 2048                # RSA key size
encrypt_key           = yes                  # Protect private key
default_md             = sha1                 # MD to use
utf8                   = yes                  # Input is UTF-8
string_mask            = utf8only             # Emit UTF-8 strings
prompt                = no                   # Don't prompt for DN
distinguished_name     = encryption_dn        # DN template
req_extensions         = encryption_reqext    # Desired extensions

[ encryption_dn ]
0.domainComponent     = "edu"
1.domainComponent     = "gxun"
organizationName      = "软件学院"
organizationalUnitName = "18信安"
commonName            = " "

[ encryption_reqext ]
keyUsage               = critical,digitalSignature,dataEncipherment,keyEncipherment
extendedKeyUsage       = OCSPSigning,serverAuth,clientAuth,emailProtection,codeSigning,timeStamping
subjectKeyIdentifier   = hash
```

6.2 创建用于数据加密的证书请求

req -new \

- config etc/encryption.conf \
- out certs/encryption.csr \
- keyout certs/encryption.key

6.3 生成用于数据加密的证书

ca \

- config etc/signing-ca.conf \
- in certs/encryption.csr \
- out certs/encryption.crt \
- extensions encryption_ext

```
管理员: 命令提示符 - openssl
openssl req -new \
  -config etc/encryption.conf \
  -out certs/encryption.csr \
  -keyout certs/encryption.key
Generating a RSA private key
.....+++++
.....+++++
writing new private key to 'certs/encryption.key'
Enter PEM pass phrase:
Verifying - Enter PEM pass phrase:

openssl ca \
  -config etc/signing-ca.conf \
  -in certs/encryption.csr \
  -out certs/encryption.crt \
  -extensions encryption_ext
Using configuration from etc/signing-ca.conf
Enter pass phrase for ./ca/signing-ca/private/signing-ca.key:
Check that the request matches the signature
Signature ok
Certificate Details:
  Serial Number: 3 (0x3)
  Validity
    Not Before: May 14 12:24:55 2020 GMT
    Not After : May 14 12:24:55 2022 GMT
  Subject:
    domainComponent       = edu
    domainComponent       = gxun
    organizationName      = \USF6F\U4EF6\U5B66\U9662
    organizationalUnitName = 18\U4FE1\U5B89
    commonName            = \UC\U\U
  X509v3 extensions:
    X509v3 Key Usage: critical
      Digital Signature, Key Encipherment, Data Encipherment
    X509v3 Basic Constraints:
      CA:FALSE
    X509v3 Extended Key Usage:
      OCSP Signing, TLS Web Server Authentication, TLS Web Client Authentication, E-mail Protection, Code Signing, Time Stamping
    X509v3 Subject Key Identifier:
      80:99:2E:07:3D:35:C5:C6:64:41:EF:7A:E4:4F:EB:B1:20:BB:C2:CF
    X509v3 Authority Key Identifier:
      keyid:CB:7D:5E:45:CE:21:2C:37:BE:19:A7:DB:BD:B4:10:DE:F6:C5:BF:9A

Certificate is to be certified until May 14 12:24:55 2022 GMT (730 days)
Sign the certificate? [y/n]:y

1 out of 1 certificate requests certified, commit? [y/n]:y
Write out database with 1 new entries
Data Base Updated
```

4、5、6 部分结果如图：

此电脑 > 本地磁盘 (F:) > pki-example-1 > certs					搜索"certs"
名称	修改日期	类型	大小		
email.crt	2020/5/14 20:18	安全证书	5 KB		
email.csr	2020/5/14 20:17	CSR 文件	2 KB		
email.key	2020/5/14 20:17	KEY 文件	2 KB		
encryption.crt	2020/5/14 20:25	安全证书	5 KB		
encryption.csr	2020/5/14 20:24	CSR 文件	2 KB		
encryption.key	2020/5/14 20:24	KEY 文件	2 KB		
identity.crt	2020/5/14 20:19	安全证书	5 KB		
identity.csr	2020/5/14 20:18	CSR 文件	2 KB		
identity.key	2020/5/14 20:18	KEY 文件	2 KB		

7. 废除第 6 步用于加密的证书并将其加入到 CRL 中

7.1 吊销用于加密的证书（查看证书序列号为 03）

ca \

```
-config etc/signing-ca.conf \  
-revoke ca/signing-ca/03.pem \  
-crl_reason superseded
```

```
OpenSSL> ca \  
> -config etc/signing-ca.conf \  
> -revoke ca/signing-ca/03.pem \  
> -crl_reason superseded  
Using configuration from etc/signing-ca.conf  
Enter pass phrase for ./ca/signing-ca/private/signing-ca.key:   
Revoking Certificate 03.  
Data Base Updated
```

7.2 创建证书撤销列表

ca -gencrl \

```
-config etc/signing-ca.conf \  
-out crl/signing-ca.crl
```

```
OpenSSL> ca -gencrl \  
> -config etc/signing-ca.conf \  
> -out crl/signing-ca.crl  
Using configuration from etc/signing-ca.conf  
Enter pass phrase for ./ca/signing-ca/private/signing-ca.key:   
crl 中新增文件 signing-ca.crl，吊销成功：
```

crl 中新增文件 signing-ca.crl，吊销成功：

证书吊销列表

常规 吊销列表

证书吊销列表信息

字段	值
版本	V2
颁发者	signingCA, 18信安, 软件学院, ...
生效日期	2020年5月14日 20:31:13
下一次更新的时间	2020年5月21日 20:31:13
签名算法	sha1RSA
签名哈希算法	sha1
授权密钥标识符	KeyID=cb7d5e45ce212c37be...
CRL 数字	01
指纹	c8d2dfc940efef58fa5ec78902...

证书吊销列表

常规 吊销列表

吊销的证书(R):

序列号	吊销日期
03	2020年5月14日 20:3...

吊销项(E)

字段	值
序列号	03
吊销日期	2020年5月14日 20:30:34
CRL 理由码	被取代 (4)