实验 1：私有 CA 证书签发的简单实现

实验要求

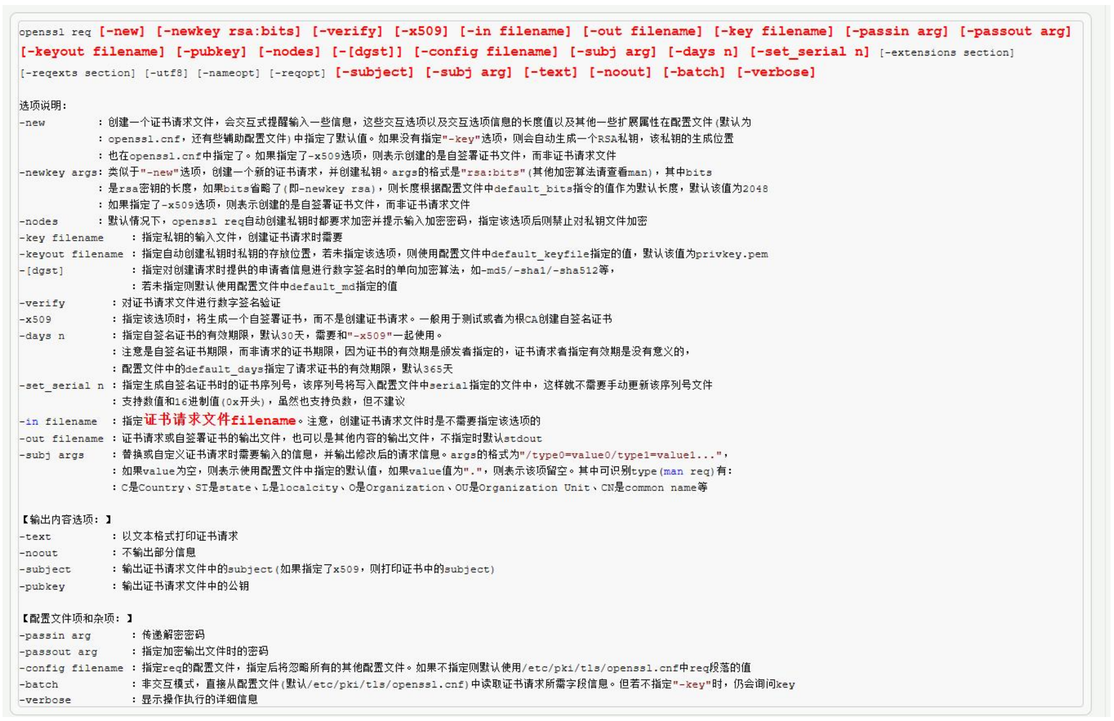
1、熟悉 Openssl 工具的使用

2、搭建私有 CA 并生成根证书

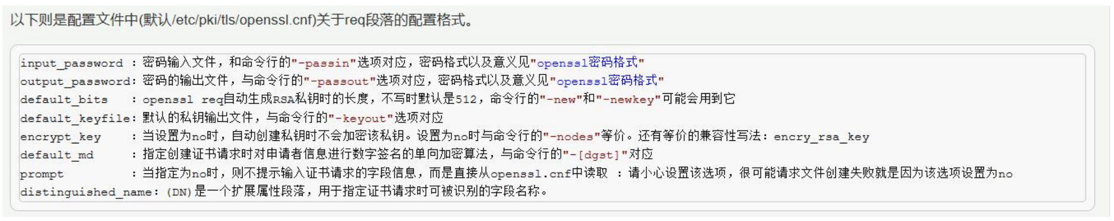
3、完成证书签发、吊销流程的简单实现

实验准备

OpenSSL req

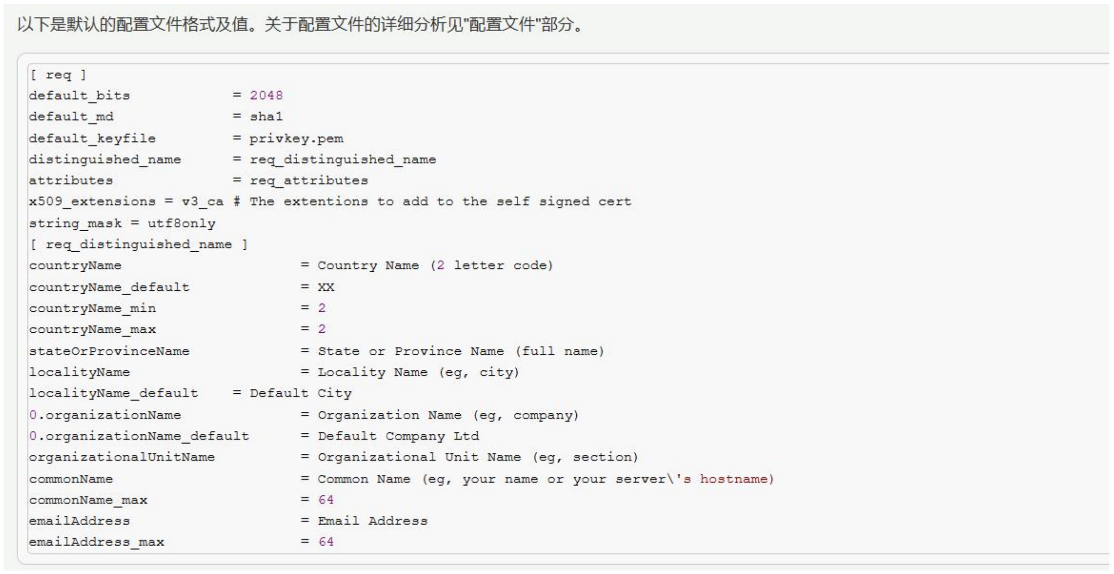


```
openssl req [-new] [-newkey rsa:bits] [-verify] [-x509] [-in filename] [-out filename] [-key filename] [-passin arg] [-passout arg]
[-keyout filename] [-pubkey] [-nodes] [-[dgst]] [-config filename] [-subj arg] [-days n] [-set_serial n] [-extensions section]
[-reqexts section] [-utf8] [-nameopt] [-reqopt] [-subject] [-subj arg] [-text] [-noout] [-batch] [-verbose]

选项说明:
-new         : 创建一个证书请求文件，会交互式提醒输入一些信息，这些交互选项以及交互选项信息的长度值以及其他一些扩展属性在配置文件(默认为
             : openssl.cnf，还有些辅助配置文件)中指定了默认值。如果没有指定"-key"选项，则会自动生成一个RSA私钥，该私钥的生成位置
             : 也在openssl.cnf中指定了。如果指定了-x509选项，则表示创建的是自签署证书文件，而非证书请求文件
-newkey args: 类似于"-new"选项，创建一个新的证书请求，并创建私钥。args的格式是"rsa:bits"(其他加密算法请查看man)，其中bits
             : 是rsa密钥的长度，如果bits省略了(即-newkey rsa)，则长度根据配置文件中default_bits指令的值作为默认长度，默认该值为2048
             : 如果指定了-x509选项，则表示创建的是自签署证书文件，而非证书请求文件
-nodes       : 默认情况下，openssl req自动创建私钥时都要求加密并提示输入加密密码，指定该选项后则禁止对私钥文件加密
-key filename     : 指定私钥的输入文件，创建证书请求时需要
-keyout filename : 指定自动创建私钥时私钥的存放位置，若未指定该选项，则使用配置文件中default_keyfile指定的值，默认该值为privkey.pem
-[dgst]           : 指定对创建请求时提供的申请者信息进行数字签名时的单向加密算法，如-md5/-sha1/-sha512等，
                  : 若未指定则默认使用配置文件中default_md指定的值
-verify         : 对证书请求文件进行数字签名验证
-x509           : 指定该选项时，将生成一个自签署证书，而不是创建证书请求。一般用于测试或者为根CA创建自签名证书
-days n         : 指定自签名证书的有效期限，默认30天，需要和"-x509"一起使用。
                : 注意是自签名证书期限，而非请求的证书期限，因为证书的有效期是颁发者指定的，证书请求者指定有效期是没有意义的，
                : 配置文件中的default_days指定了请求证书的有效期限，默认365天
-set_serial n : 指定生成自签名证书时的证书序列号，该序列号将写入配置文件中serial指定的文件中，这样就不需要手动更新该序列号文件
                : 支持数值和16进制值(0x开头)，虽然也支持负数，但不建议
-in filename    : 指定证书请求文件filename。注意，创建证书请求文件时是不需要指定该选项的
-out filename : 证书请求或自签署证书的输出文件，也可以是其他内容的输出文件，不指定时默认stdout
-subj args      : 替换或自定义证书请求时需要输入的信息，并输出修改后的请求信息。args的格式为"/type0=value0/type1=value1...",
                : 如果value为空，则表示使用配置文件中指定的默认值，如果value值为"."，则表示该项留空。其中可识别type(man req)有:
                : C是Country、ST是state、L是localcity、O是Organization、OU是Organization Unit、CN是common name等

【输出内容选项：】
-text           : 以文本格式打印证书请求
-noout          : 不输出部分信息
-subject        : 输出证书请求文件中的subject(如果指定了x509，则打印证书中的subject)
-pubkey         : 输出证书请求文件中的公钥

【配置文件项和杂项：】
-passin arg       : 传递解密密码
-passout arg      : 指定加密输出文件时的密码
-config filename : 指定req的配置文件，指定后将忽略所有的其他配置文件。如果不指定则默认使用/etc/pki/tls/openssl.cnf中req段落的值
-batch            : 非交互模式，直接从配置文件(默认/etc/pki/tls/openssl.cnf)中读取证书请求所需字段信息。但若不指定"-key"时，仍会询问key
-verbose          : 显示操作执行的详细信息
```



以下则是配置文件中(默认/etc/pki/tls/openssl.cnf)关于req段落的配置格式。

```
input_password : 密码输入文件，和命令行的"-passin"选项对应，密码格式以及意义见"openssl密码格式"
output_password: 密码的输出文件，与命令行的"-passout"选项对应，密码格式以及意义见"openssl密码格式"
default_bits    : openssl req自动生成RSA私钥时的长度，不写时默认是512，命令行的"-new"和"-newkey"可能会用到它
default_keyfile: 默认的私钥输出文件，与命令行的"-keyout"选项对应
encrypt_key     : 当设置为no时，自动创建私钥时不会加密该私钥。设置为no时与命令行的"-nodes"等价。还有等价的兼容性写法：encry_rsa_key
default_md      : 指定创建证书请求时对申请者信息进行数字签名的单向加密算法，与命令行的"-[dgst]"对应
prompt          : 当指定为no时，则不提示输入证书请求的字段信息，而是直接从openssl.cnf中读取 ：请小心设置该选项，很可能请求文件创建失败就是因为该选项设置为no
distinguished_name: (DN)是一个扩展属性段落，用于指定证书请求时可被识别的字段名称。
```


以下是默认的配置文件格式及值。关于配置文件的详细分析见"配置文件"部分。

```
[ req ]
default_bits            = 2048
default_md              = sha1
default_keyfile         = privkey.pem
distinguished_name      = req_distinguished_name
attributes              = req_attributes
x509_extensions = v3_ca # The extentions to add to the self signed cert
string_mask = utf8only
[ req_distinguished_name ]
countryName                     = Country Name (2 letter code)
countryName_default             = XX
countryName_min                 = 2
countryName_max                 = 2
stateOrProvinceName             = State or Province Name (full name)
localityName                    = Locality Name (eg, city)
localityName_default    = Default City
0.organizationName              = Organization Name (eg, company)
0.organizationName_default      = Default Company Ltd
organizationalUnitName          = Organizational Unit Name (eg, section)
commonName                      = Common Name (eg, your name or your server\'s hostname)
commonName_max                  = 64
emailAddress                    = Email Address
emailAddress_max                = 64
```

参考网址：

1.(22 条消息) 基于 openssl 工具完成自建 CA 以及为 server,client 颁发证书_tutu-hu 的博客-CSDN 博客_ca.crt server.crt 自建证书

2.openssl req(生成证书请求和自建 CA) - 骏马金龙 - 博客园 (cnblogs.com)

3.(22 条消息) Linux 实现搭建私有 CA 服务器和证书申请颁发吊销_白-胖-子的博客-CSDN 博客_linux policy/match

实验环境

Ubuntu 20.04+OpenSSL 1.1

实验内容

1、搭建私有 CA

（1）创建私有 CA 所需要的文件目录，保存 CA 的相关信息

mkdir myCA　　　　　　　　　//创建 CA 根文件夹

cd myCA　　　　　　　　　　//进入 CA 根文件夹

mkdir newcerts private conf　　//创建三个文件夹，用来存放新发放证书、私钥和配置文件

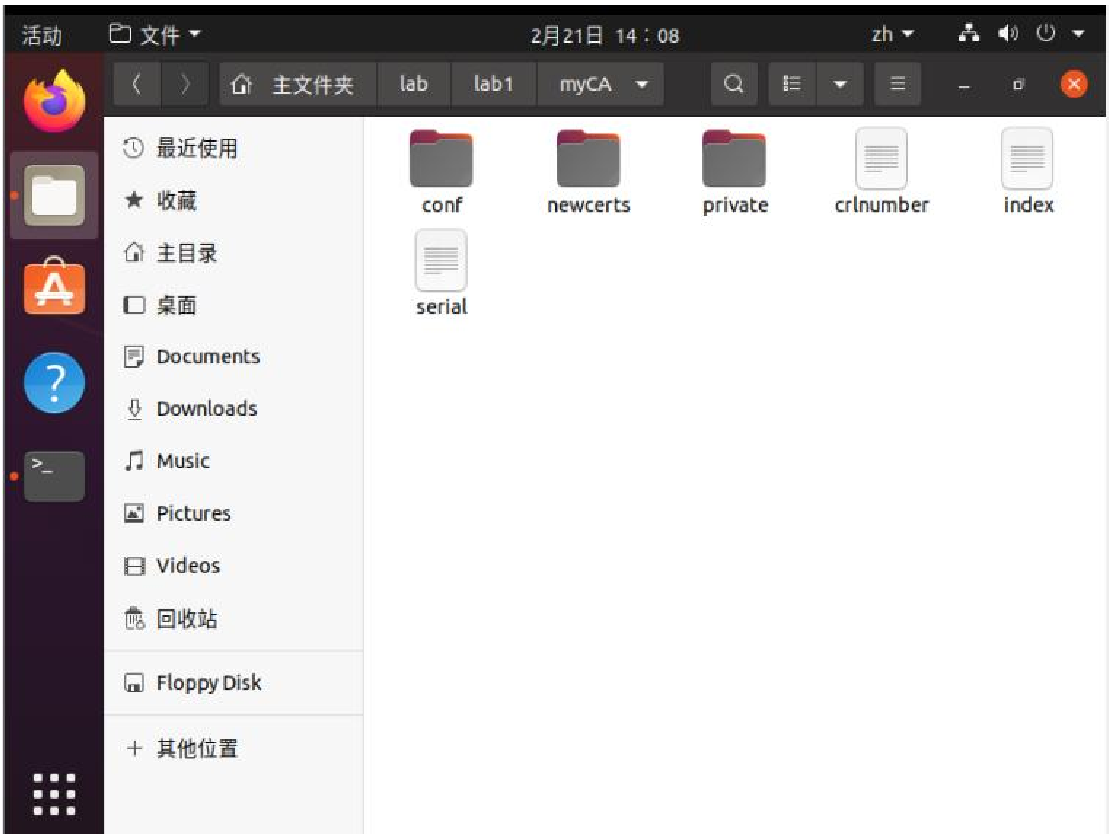
chmod g-rwx,o-rwx private　　//设置 private 文件夹的操作权限

touch index crlnumber　　　　//创建证书信息数据库、crl 编号列表

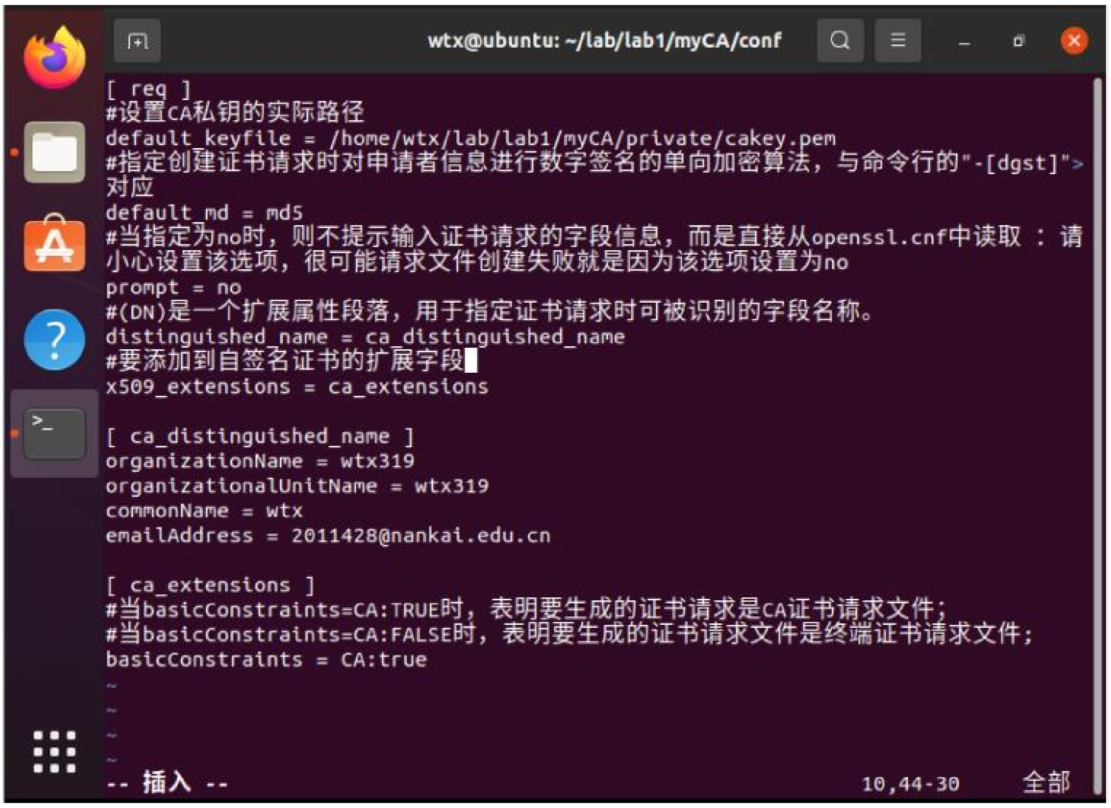
echo 01 > serial　　　　　　　//初始化证书的序列号

echo 01 > crlnumber　　　　　　//初始化吊销证书列表序号

结果如下：



（2）创建生成 CA 自签名证书的配置文件



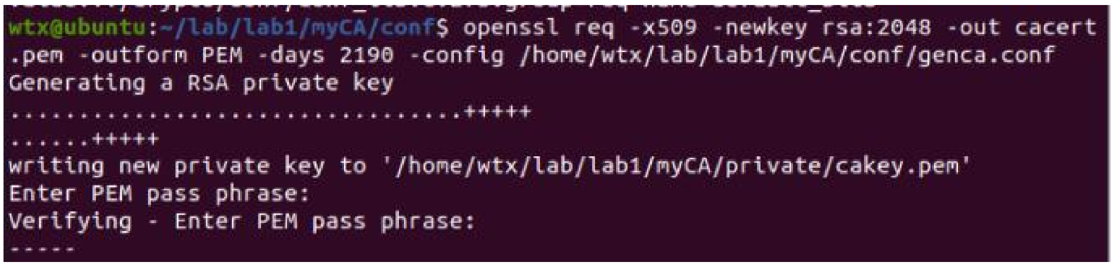
```
[ req ]
#设置CA私钥的实际路径
default_keyfile = /home/wtx/lab/lab1/myCA/private/cakey.pem
#指定创建证书请求时对申请者信息进行数字签名的单向加密算法，与命令行的"-[dgst]">
对应
default_md = md5
#当指定为no时，则不提示输入证书请求的字段信息，而是直接从openssl.cnf中读取 ： 请
小心设置该选项，很可能请求文件创建失败就是因为该选项设置为no
prompt = no
#(DN)是一个扩展属性段落，用于指定证书请求时可被识别的字段名称。
distinguished_name = ca_distinguished_name
#要添加到自签名证书的扩展字段
x509_extensions = ca_extensions

[ ca_distinguished_name ]
organizationName = wtx319
organizationalUnitName = wtx319
commonName = wtx
emailAddress = 2011428@nankai.edu.cn

[ ca_extensions ]
#当basicConstraints=CA:TRUE时，表明要生成的证书请求是CA证书请求文件；
#当basicConstraints=CA:FALSE时，表明要生成的证书请求文件是终端证书请求文件；
basicConstraints = CA:true
~
~
~
~
-- 插入 --                                                  10,44-30      全部
```

（3）生成私有 CA 的私钥和自签名证书（根证书）

openssl req -x509 -newkey rsa:2048 -out /home/wtx/lab/lab1/myCA/newcarts/cacert.pem -outform PEM -days 2190 -config /home/wtx/lab/lab1/myCA/conf/genca.conf

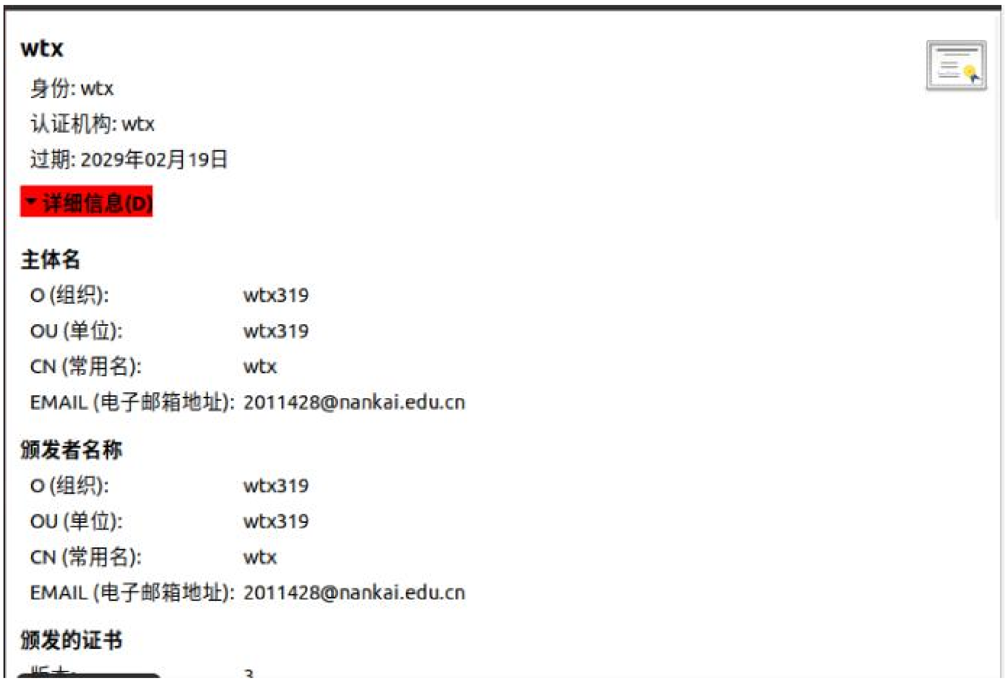
//生成 x509 的 CA 证书，过程中需要输入 CA 私钥的保护密码（cakey），请牢记。

//CA 会按照 genca.conf 文件中配置的规则自签名生成证书



```
wtx@ubuntu:~/lab/lab1/myCA/conf$ openssl req -x509 -newkey rsa:2048 -out cacert
.pem -outform PEM -days 2190 -config /home/wtx/lab/lab1/myCA/conf/genca.conf
Generating a RSA private key
.................................+++++
......+++++
writing new private key to '/home/wtx/lab/lab1/myCA/private/cakey.pem'
Enter PEM pass phrase:
Verifying - Enter PEM pass phrase:
-----
```

在 newcarts 文件夹下出现了 cacert.pem 文件。



wtx
身份: wtx
认证机构: wtx
过期: 2029年02月19日
▾ 详细信息(D)

主体名

O (组织):	wtx319
OU (单位):	wtx319
CN (常用名):	wtx
EMAIL (电子邮箱地址):	2011428@nankai.edu.cn

颁发者名称

O (组织):	wtx319
OU (单位):	wtx319
CN (常用名):	wtx
EMAIL (电子邮箱地址):	2011428@nankai.edu.cn

颁发的证书

2、私有 CA 为服务器签发证书

（1）创建用来为其他请求签发证书的配置文件

reca.conf

```
####################################
[ ca ]
default_ca          = testca                            #  按需修改

[ testca ]                                                     #  按需修改
dir                     = /home/wtx/lab/lab1/myCA            # top dir，按实际给出
database              = $dir/index                # index file.
new_certs_dir         = $dir/newcerts                 # new certs dir

certificate             = $dir/private/cacert.pem                 # The CA cert
serial                       = $dir/serial                            # serial no file
private_key           = $dir/private/cakey.pem      # CA private key
RANDFILE              = $dir/private/.rand           # random number file

default_days      = 365                                   # how long to certify for
default_crl_days= 30                                      # how long before next CRL
default_md         = md5                                   # message digest method to use
unique_subject = no                                       # Set to 'no' to allow creation of
                                                                 # several ctificates with same subject.
policy              = policy_any                          # default policy

[ policy_any ]
countryName                      = optional
stateOrProvinceName         = optional
localityName                    = optional
organizationName              = optional
organizationalUnitName    = optional
commonName                         = supplied
emailAddress                    = optional

##########################################
```

（2）模拟服务器，生成私钥与证书申请的请求文件

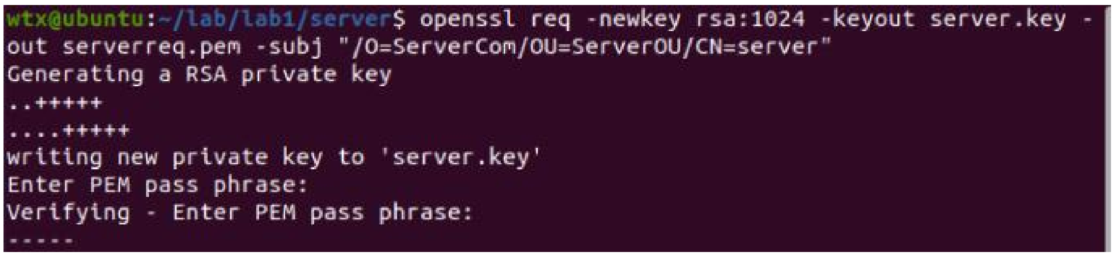
（3）CA 根据服务器的证书请求文件生成证书并将其返回给服务器

在与 myCA 同级，创建文件夹 server。

生成 server 的私钥 server.key 及证书申请的请求文件 serverreq.pem：

openssl req -newkey rsa:1024 -keyout server.key -out serverreq.pem -subj "/O=ServerCom/OU=ServerOU/CN=server"

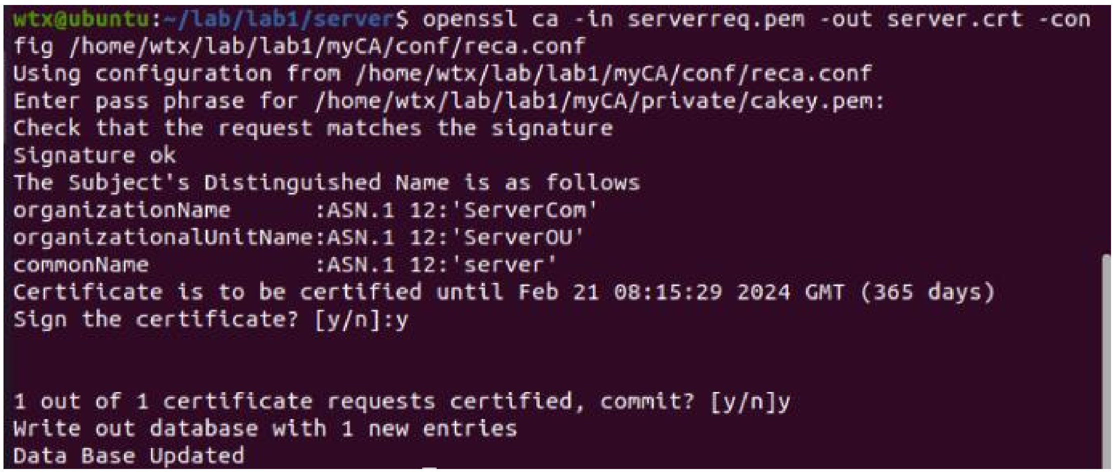
私钥为：server



```
wtx@ubuntu:~/lab/lab1/server$ openssl req -newkey rsa:1024 -keyout server.key -
out serverreq.pem -subj "/O=ServerCom/OU=ServerOU/CN=server"
Generating a RSA private key
..+++++
....+++++
writing new private key to 'server.key'
Enter PEM pass phrase:
Verifying - Enter PEM pass phrase:
-----
```

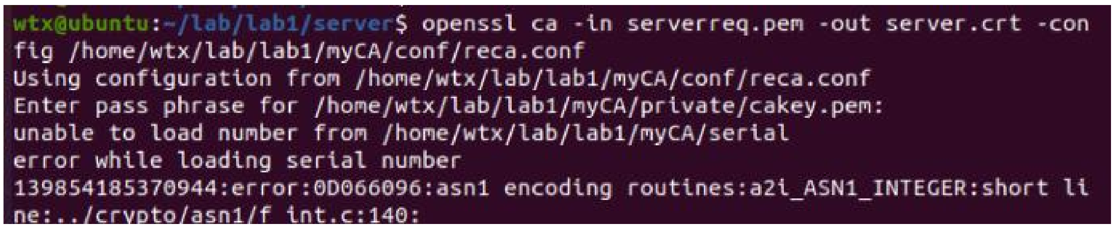
提交 serverreq.pem 向 CA 申请证书并生成证书 server.crt：

openssl ca -in serverreq.pem -out server.crt -config /home/wtx/lab/lab1/myCA/conf/reca.conf



```
wtx@ubuntu:~/lab/lab1/server$ openssl ca -in serverreq.pem -out server.crt -con
fig /home/wtx/lab/lab1/myCA/conf/reca.conf
Using configuration from /home/wtx/lab/lab1/myCA/conf/reca.conf
Enter pass phrase for /home/wtx/lab/lab1/myCA/private/cakey.pem:
Check that the request matches the signature
Signature ok
The Subject's Distinguished Name is as follows
organizationName      :ASN.1 12:'ServerCom'
organizationalUnitName:ASN.1 12:'ServerOU'
commonName            :ASN.1 12:'server'
Certificate is to be certified until Feb 21 08:15:29 2024 GMT (365 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
```

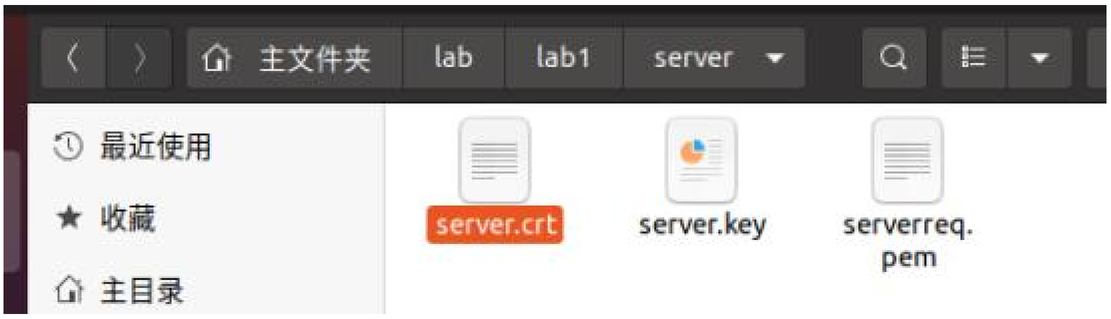
此时遇到报错：



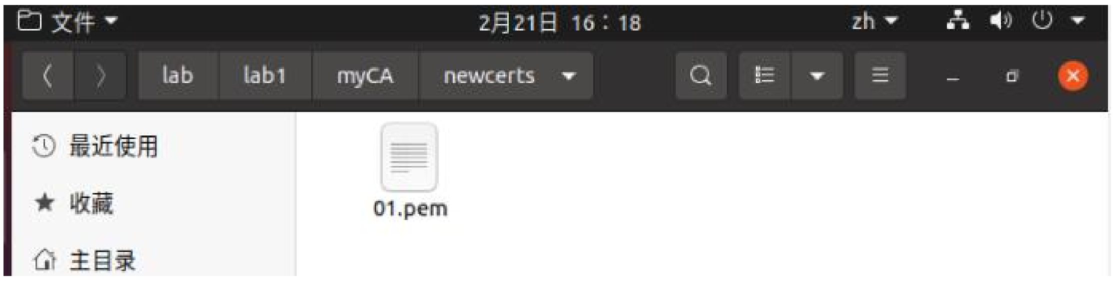
```
wtx@ubuntu:~/lab/lab1/server$ openssl ca -in serverreq.pem -out server.crt -con
fig /home/wtx/lab/lab1/myCA/conf/reca.conf
Using configuration from /home/wtx/lab/lab1/myCA/conf/reca.conf
Enter pass phrase for /home/wtx/lab/lab1/myCA/private/cakey.pem:
unable to load number from /home/wtx/lab/lab1/myCA/serial
error while loading serial number
139854185370944:error:0D066096:asn1 encoding routines:a2i_ASN1_INTEGER:short li
ne:../crypto/asn1/f_int.c:140:
```

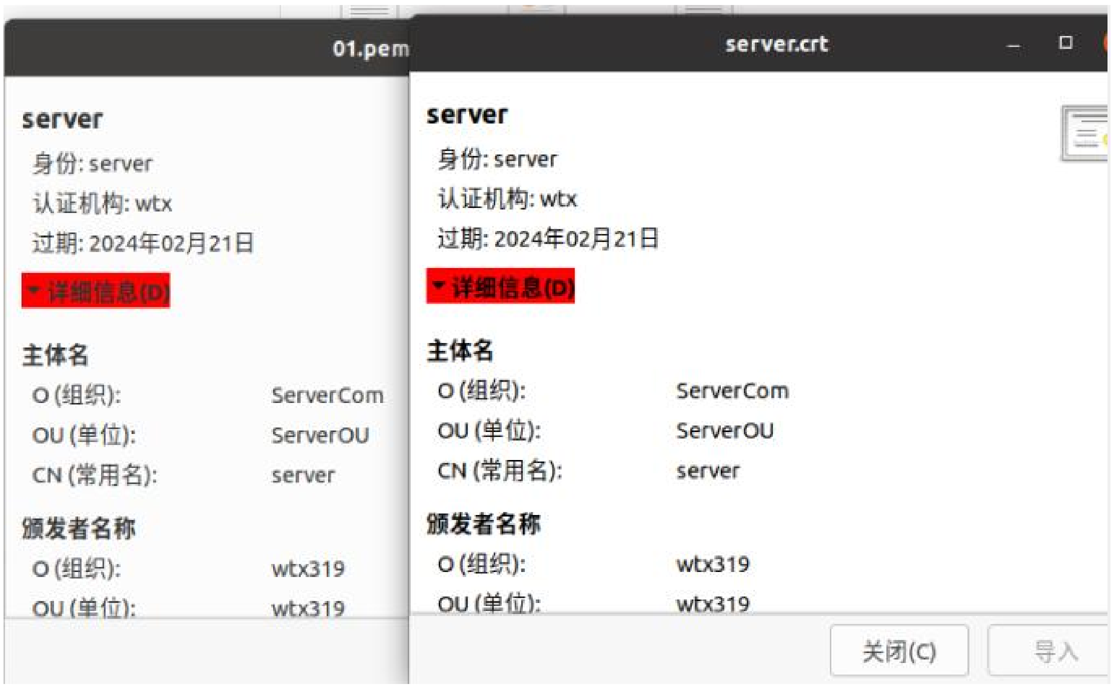
解决方案：echo "01" > /home/wtx/lab/lab1/myCA/serial

结果：



同时，在 CA 目录下 newcerts 目录下也生成了该证书的备份：01.pem，这和备份的内容和生成的证书 server.crt 完全一致。




01.pem

server
身份: server
认证机构: wtx
过期: 2024年02月21日
▾ 详细信息(D)
主体名
O (组织): ServerCom
OU (单位): ServerOU
CN (常用名): server
颁发者名称
O (组织): wtx319
OU (单位): wtx319

server.crt

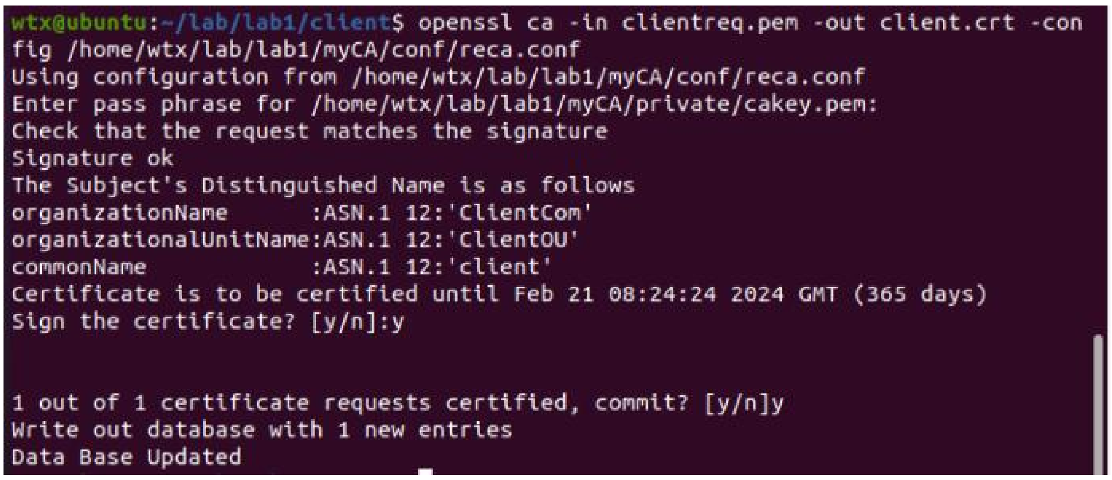
server
身份: server
认证机构: wtx
过期: 2024年02月21日
▾ 详细信息(D)
主体名
O (组织): ServerCom
OU (单位): ServerOU
CN (常用名): server
颁发者名称
O (组织): wtx319
OU (单位): wtx319

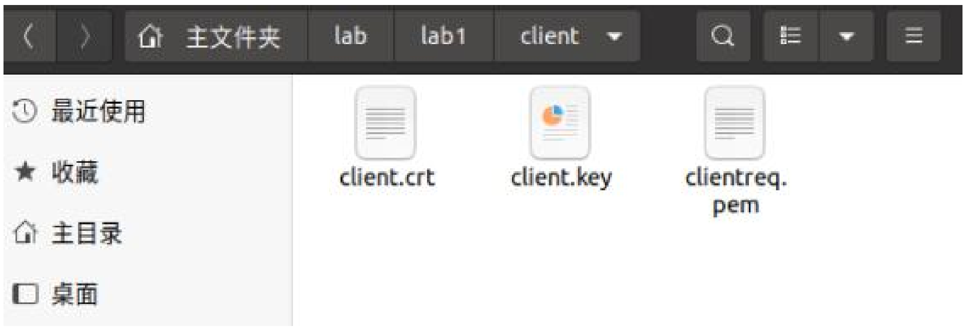
3、私有 CA 为客户端签发证书

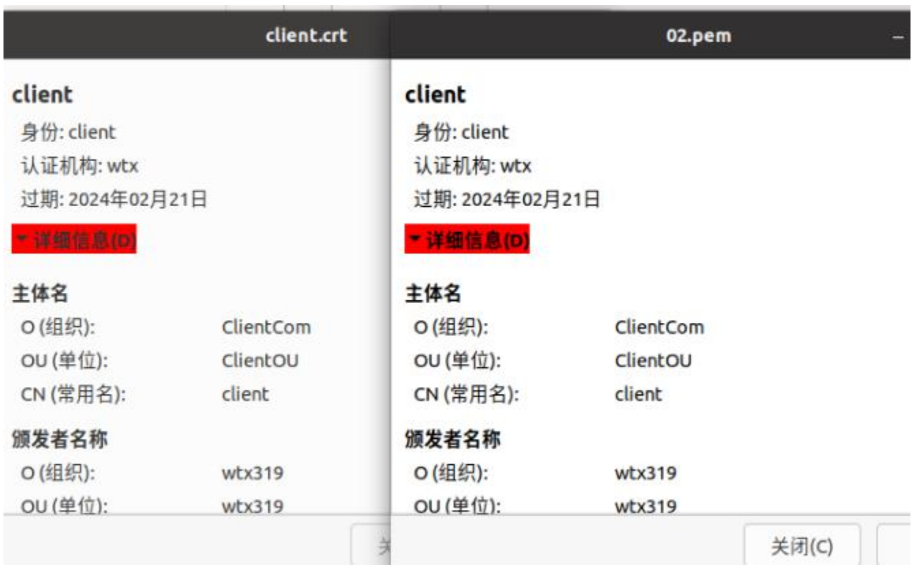
整体步骤同 server 类似。

openssl req -newkey rsa:1024 -keyout client.key -out clientreq.pem -subj "/O=ClientCom/OU=ClientOU/CN=client"（密钥：client）

openssl ca -in clientreq.pem -out client.crt -config /home/wtx/lab/lab1/myCA/conf/reca.conf


```
wtx@ubuntu:~/lab/lab1/client$ openssl ca -in clientreq.pem -out client.crt -con
fig /home/wtx/lab/lab1/myCA/conf/reca.conf
Using configuration from /home/wtx/lab/lab1/myCA/conf/reca.conf
Enter pass phrase for /home/wtx/lab/lab1/myCA/private/cakey.pem:
Check that the request matches the signature
Signature ok
The Subject's Distinguished Name is as follows
organizationName      :ASN.1 12:'ClientCom'
organizationalUnitName:ASN.1 12:'ClientOU'
commonName            :ASN.1 12:'client'
Certificate is to be certified until Feb 21 08:24:24 2024 GMT (365 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
```





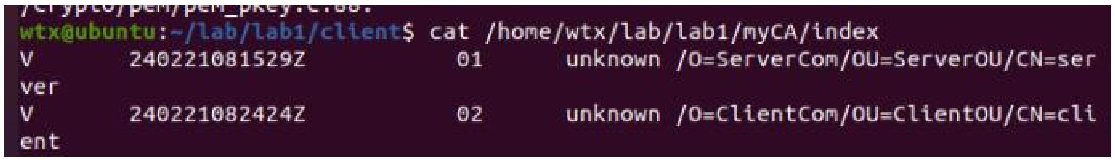
client
身份: client
认证机构: wtx
过期: 2024年02月21日
详细信息(D)
主体名
O (组织): ClientCom
OU (单位): ClientOU
CN (常用名): client
颁发者名称
O (组织): wtx319
OU (单位): wtx319

client
身份: client
认证机构: wtx
过期: 2024年02月21日
详细信息(D)
主体名
O (组织): ClientCom
OU (单位): ClientOU
CN (常用名): client
颁发者名称
O (组织): wtx319
OU (单位): wtx319

4、CA 吊销用户证书

（1）找到证书对应的编号

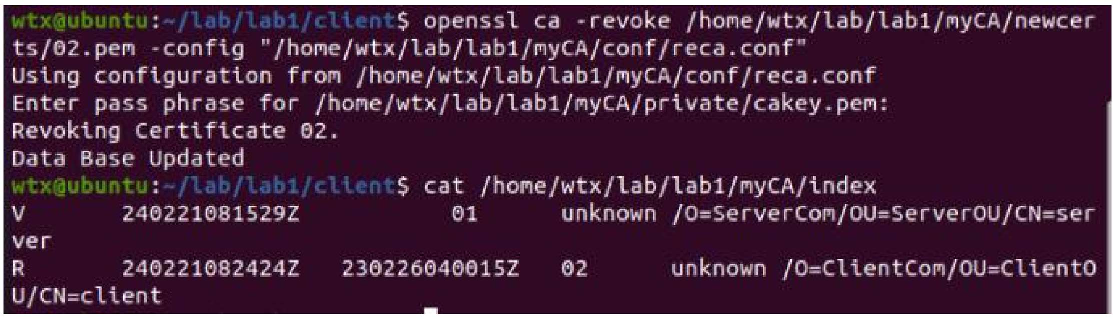
cat /home/wtx/lab/lab1/myCA/index



```
wtx@ubuntu:~/lab/lab1/client$ cat /home/wtx/lab/lab1/myCA/index
V       240221081529Z           01      unknown /O=ServerCom/OU=ServerOU/CN=ser
ver
V       240221082424Z           02      unknown /O=ClientCom/OU=ClientOU/CN=cli
ent
```

（2）在存放证书的文件夹下找到编号对应的证书，对其完成吊销

openssl ca -revoke /home/wtx/lab/lab1/myCA/newcerts/02.pem -config "/home/wtx/lab/lab1/myCA/conf/reca.conf"

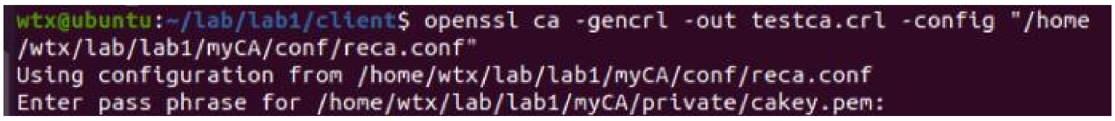


```
wtx@ubuntu:~/lab/lab1/client$ openssl ca -revoke /home/wtx/lab/lab1/myCA/newcer
ts/02.pem -config "/home/wtx/lab/lab1/myCA/conf/reca.conf"
Using configuration from /home/wtx/lab/lab1/myCA/conf/reca.conf
Enter pass phrase for /home/wtx/lab/lab1/myCA/private/cakey.pem:
Revoking Certificate 02.
Data Base Updated
wtx@ubuntu:~/lab/lab1/client$ cat /home/wtx/lab/lab1/myCA/index
V       240221081529Z           01      unknown /O=ServerCom/OU=ServerOU/CN=ser
ver
R       240221082424Z   230226040015Z   02      unknown /O=ClientCom/OU=ClientO
U/CN=client
```

（3）更新吊销证书列表

生成证书吊销列表文件（CRL）

openssl ca -gencrl -out testca.crl -config "/home/wtx/lab/lab1/myCA/conf/reca.conf"



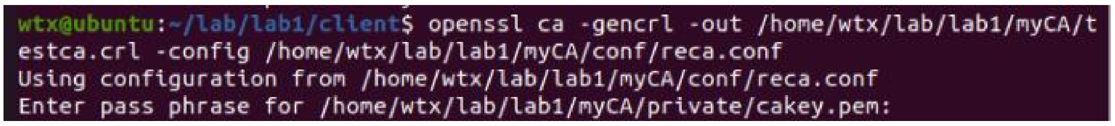
```
wtx@ubuntu:~/lab/lab1/client$ openssl ca -gencrl -out testca.crl -config "/home
/wtx/lab/lab1/myCA/conf/reca.conf"
Using configuration from /home/wtx/lab/lab1/myCA/conf/reca.conf
Enter pass phrase for /home/wtx/lab/lab1/myCA/private/cakey.pem:
```

指定第一个吊销证书的编号,注意：第一次更新证书吊销列表前，才需要执行

echo 01 > /home/wtx/lab/lab1/myCA/crlnumber

更新证书吊销列表

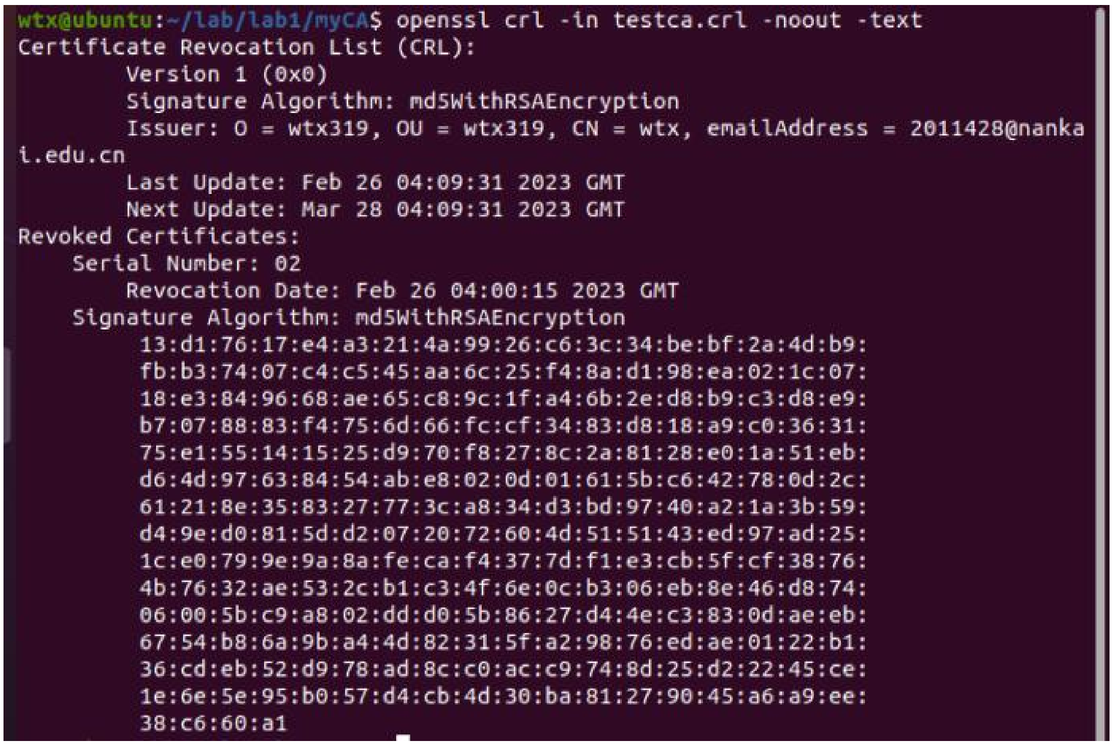
openssl ca -gencrl -out /home/wtx/lab/lab1/myCA/testca.crl -config /home/wtx/lab/lab1/myCA/conf/reca.conf



```
wtx@ubuntu:~/lab/lab1/client$ openssl ca -gencrl -out /home/wtx/lab/lab1/myCA/t
estca.crl -config /home/wtx/lab/lab1/myCA/conf/reca.conf
Using configuration from /home/wtx/lab/lab1/myCA/conf/reca.conf
Enter pass phrase for /home/wtx/lab/lab1/myCA/private/cakey.pem:
```

查看吊销 crl 文件：

openssl crl -in testca.crl -noout -text



```
wtx@ubuntu:~/lab/lab1/myCA$ openssl crl -in testca.crl -noout -text
Certificate Revocation List (CRL):
        Version 1 (0x0)
        Signature Algorithm: md5WithRSAEncryption
        Issuer: O = wtx319, OU = wtx319, CN = wtx, emailAddress = 2011428@nanka
i.edu.cn
        Last Update: Feb 26 04:09:31 2023 GMT
        Next Update: Mar 28 04:09:31 2023 GMT
Revoked Certificates:
    Serial Number: 02
        Revocation Date: Feb 26 04:00:15 2023 GMT
    Signature Algorithm: md5WithRSAEncryption
         13:d1:76:17:e4:a3:21:4a:99:26:c6:3c:34:be:bf:2a:4d:b9:
         fb:b3:74:07:c4:c5:45:aa:6c:25:f4:8a:d1:98:ea:02:1c:07:
         18:e3:84:96:68:ae:65:c8:9c:1f:a4:6b:2e:d8:b9:c3:d8:e9:
         b7:07:88:83:f4:75:6d:66:fc:cf:34:83:d8:18:a9:c0:36:31:
         75:e1:55:14:15:25:d9:70:f8:27:8c:2a:81:28:e0:1a:51:eb:
         d6:4d:97:63:84:54:ab:e8:02:0d:01:61:5b:c6:42:78:0d:2c:
         61:21:8e:35:83:27:77:3c:a8:34:d3:bd:97:40:a2:1a:3b:59:
         d4:9e:d0:81:5d:d2:07:20:72:60:4d:51:51:43:ed:97:ad:25:
         1c:e0:79:9e:9a:8a:fe:ca:f4:37:7d:f1:e3:cb:5f:cf:38:76:
         4b:76:32:ae:53:2c:b1:c3:4f:6e:0c:b3:06:eb:8e:46:d8:74:
         06:00:5b:c9:a8:02:dd:d0:5b:86:27:d4:4e:c3:83:0d:ae:eb:
         67:54:b8:6a:9b:a4:4d:82:31:5f:a2:98:76:ed:ae:01:22:b1:
         36:cd:eb:52:d9:78:ad:8c:c0:ac:c9:74:8d:25:d2:22:45:ce:
         1e:6e:5e:95:b0:57:d4:cb:4d:30:ba:81:27:90:45:a6:a9:ee:
         38:c6:60:a1
```

实验结果

•CA 如何验证证书的有效性？

证书的签发过程：

1. 服务方 S 向第三方机构 CA 提交公钥、组织信息、个人信息(域名)等信息并申请认证；

2. CA 通过线上、线下等多种手段验证申请者提供信息的真实性，如组织是否存在、企业是否合法，是否拥有域名的所有权等；

3. 如信息审核通过，CA 会向申请者签发认证文件-证书。证书包含以下信息：申请者公钥、申请者的组织信息和个人信息、签发机构 CA 的信息、有效时间、证书序列号等信息的明文，同时包含一个签名；签名的产生算法：首先，使用散列函数计算公开的明文信息的信息摘

要，然后，采用 CA 的私钥对信息摘要进行加密，密文即签名；

4. 客户端 C 向服务器 S 发出请求时，S 返回证书文件；

5. 客户端 C 读取证书中的相关的明文信息，采用相同的散列函数计算得到信息摘要，然后，利用对应 CA 的公钥解密签名数据，对比证书的信息摘要，如果一致，则可以确认证书的合法性，即公钥合法；

6. 客户端然后验证证书相关的域名信息、有效时间等信息；

7. 客户端会内置信任 CA 的证书信息(包含公钥)，如果CA不被信任，则找不到对应 CA 的证书，证书也会被判定非法。

在这个过程注意几点：

1.申请证书不需要提供私钥，确保私钥永远只能服务器掌握；

2.证书的合法性仍然依赖于非对称加密算法，证书主要是增加了服务器信息以及签名；

3.内置 CA 对应的证书称为根证书，颁发者和使用者相同，自己为自己签名，即自签名证书；

4.证书=公钥+申请者与颁发者信息+签名；

•需要考虑到哪些方面？

1. 证书是否被吊销及证书是否失效

2. 用户信息

3. 通过 hash 值判断证书是否被篡改