openssl 命令行使用简介

2017-08-21

1. 常用命令

```
openssl -help


# 支持的标准命令，即伪命令
Standard commands
asn1parse         ca                ciphers           cms
crl               crl2pkcs7         dgst              dh
dhparam           dsa               dsaparam          ec
ecparam           enc               engine            errstr
gendh             gendsa            genpkey           genrsa
nseq              ocsp              passwd            pkcs12
pkcs7             pkcs8             pkey              pkeyparam
pkeyutl           prime             rand              req
rsa               rsautl            s_client          s_server
s_time            sess_id           smime             speed
spkac             ts                verify            version
x509


# 指定"dgst"命令时即单向加密支持的算法，实际上支持更多的算法，具体见 dgst 命令
Message Digest commands (see the `dgst' command for more details)
md2               md4               md5               rmd160
sha               sha1



# 指定对称加密"enc"时支持的对称加密算法
Cipher commands (see the `enc' command for more details)
aes-128-cbc       aes-128-ecb       aes-192-cbc       aes-192-ecb
```

```
aes-256-cbc       aes-256-ecb       base64            bf
bf-cbc            bf-cfb            bf-ecb            bf-ofb
camellia-128-cbc  camellia-128-ecb  camellia-192-cbc  camellia-192-ecb
camellia-256-cbc  camellia-256-ecb  cast              cast-cbc
cast5-cbc         cast5-cfb         cast5-ecb         cast5-ofb
des               des-cbc           des-cfb           des-ecb
des-ede           des-ede-cbc       des-ede-cfb       des-ede-ofb
des-ede3          des-ede3-cbc      des-ede3-cfb      des-ede3-ofb
des-ofb           des3              desx              idea
idea-cbc          idea-cfb          idea-ecb          idea-ofb
rc2               rc2-40-cbc        rc2-64-cbc        rc2-cbc
rc2-cfb           rc2-ecb           rc2-ofb           rc4
rc4-40            seed              seed-cbc          seed-cfb
seed-ecb          seed-ofb          zlib
```

2. 公私钥处理

1. 生成 RSA 私钥

   ```
   openssl genrsa -out rsa_key.pem 2048
   ```

2. 使用私钥生成公钥

   ```
   openssl rsa -in rsa_key.pem -pubout -out rsa_pub.pem
   ```

3. 加密私钥
4. 交互方式输入密码

   ```
   openssl rsa -in rsa_key.pem -inform PEM -outform PEM -out rsa_key_crypt.pem -aes256 -passout stdin
   ```

   命令行直接指定密码

   ```
   openssl rsa -in rsa_key.pem -inform PEM -outform PEM -out rsa_key_crypt.pem -aes256 -passout pass:111111
   ```

5. 移除私钥密码

   ```
   openssl rsa -in rsa_key_crypt.pem -out rsa_key.pem
   openssl rsa -in rsa_key_crypt.pem -out rsa_key.pem -passin pass:111111
   ```

6. 将私钥转换成 PKCS8 格式

   ```
   openssl pkcs8 -topk8 -inform PEM -in rsa_key.pem -outform PEM -nocrypt -out rsa_key.pk8.pem
   ```

7. 检查私钥完整性

   ```
   openssl rsa -in rsa_key.pem -check
   ```

3. 证书生成

3.1 自签证书

创建 CA (Certificate Authority)，并生成自签证书。

```
#创建目录结构

demoCA\

    --users\              空目录，用于存放用户的私钥、CSR、证书

    --ca\                 空目录，用于存放根 CA

    --newcerts\           空目录，存放新证书

    --index.txt           空文本文件

    --index.txt.attr      空文本文件

    --serial              文本文件，输入 01，保存


#创建 CA，生成 CA 的私钥和 CA 的自签证书。

D:\Pros\OpenSSL-Win64\bin>openssl req -new -x509 -keyout demoCA\ca\ca.key -out demoCA\ca\ca.pem.crt -days 365 -config openssl.cfg

Generating a 2048 bit RSA private key

......+++

...............................+++

writing new private key to 'ca.key'

Enter PEM pass phrase:

Verifying - Enter PEM pass phrase:

-----

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]:Beijing
Locality Name (eg, city) []:Beijing
Organization Name (eg, company) [Internet Widgits Pty Ltd]:TEST
Organizational Unit Name (eg, section) []:testCA
Common Name (e.g. server FQDN or YOUR name) []:TestCA
Email Address []:test@ca.com
```

3.2 使用自建 CA 签发证书

使用上面生成的 CA 作为 Root CA（Root Certificate Authority）根证书机构，自建了数字证书注册中心 RA (Registration Authority)，为申请者签发证书。

`openssl.cfg` 文件下载 [openssl]

```
#为申请用户 server 创建私钥
D:\Pros\OpenSSL-Win64\bin>openssl genrsa -aes256 -out demoCA\users\server.key 2048
Generating RSA private key, 2048 bit long modulus
....................................................................................+++
...............................+++
e is 65537 (0x010001)
Enter pass phrase for server.key:
Verifying - Enter pass phrase for server.key:


#创建 CSR 文件，注意 CSR 中的红色部分 Organization Name，必须与 CA 的保持一致
D:\Pros\OpenSSL-Win64\bin>openssl req -new -key demoCA\users\server.key -out demoCA\users\server.csr -config openssl.cfg
Enter pass phrase for server.key:
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.
```

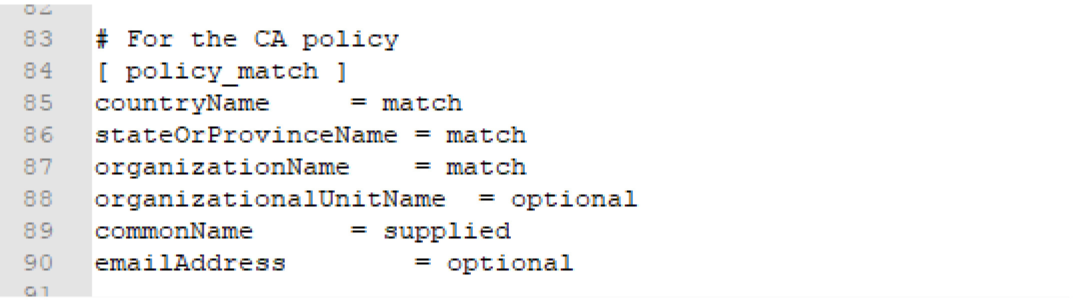
```
There are quite a few fields but you can leave some blank
For some fields there will be a default value,
If you enter '.', the field will be left blank.
-----
Country Name (2 letter code) [AU]:CN
State or Province Name (full name) [Some-State]:Beijing
Locality Name (eg, city) []:Beijing
Organization Name (eg, company) [Internet Widgits Pty Ltd]:TEST
Organizational Unit Name (eg, section) []:UnitName
Common Name (e.g. server FQDN or YOUR name) []:www.server.com
Email Address []:test@server.com

Please enter the following 'extra' attributes
to be sent with your certificate request
A challenge password []:
An optional company name []:

#查看 CSR 文件
D:\Pros\OpenSSL-Win64\bin>type demoCA\users\server.csr
-----BEGIN CERTIFICATE REQUEST-----
MIIC0jCCAboCAQAwgYwxCzAJBgNVBAYTAkNOMRAwDgYDVQQIDAdCZWlqaW5nMRAw
DgYDVQQHDAdCZWlqaW5nMQ0wCwYDVQQKDARURVNUMREwDwYDVQQLDAhVbml0TmFt
ZTEXMBUGA1UEAwwOd3d3LnNlcnZlci5jb20xHjAcBgkqhkiG9w0BCQEWD3Rlc3RA
c2VydmVyLmNvbTCCASIwDQYJKoZIhvcNAQEBBQADggEPADCCAQoCggEBAMbkt0bx
hRben8rFc6nHIhuiL0etYf9SbhmFRdnfn+n1YSqoXx/Ma8dTEAHlm93rh+QZy+JT
qsQQrVobGjDTHMMl9P6iveBVbRkWW0pogn+JDpBjg/iY25ubWvTpnK+10StaPwvj
jDSB6VyTgHtPlrAG8L61aDXzoaVvt+ASpe3x8uK37ACW4KQDSbfWi844cLp2MVu/
MZpwJYxjjebT/cnYviZ4+8rLrtnphS4c11drrSHVrFoHNE6dz18f2POLqLYE/4cf
OldPaEcPoPfU6Xub78n6d+71DbtB5NXV/r6Bp0XVbv329njh8jsYWlPzSMaR45yG
gM1XIiHlxpNn4W0CAwEAAaAAMA0GCSqGSIb3DQEBCwUAA4IBAQCfnxDBR4DAGE77
o9PaDVG0Q8yoCKDKbN+1FhgAhS/rKIEhVkOWYSlbvby1BsikYKxuqIQLN9Zvf1ed
qdNZCziqfGo8PVd7JKP+/+mLPk1tvz2qmIcMGMdIuKNmCfbIc5vEFuATQV3GkBsx
WRxhyjJKKXPU4SwSH+YL/27Ch9OIqLIkxIshOjuyOT/Q4W1IZyF17s6qjpSKFi1e
```

```
3xVOmpaWo4u9kXOxEiYeLjj1VUcusIZtAohE8sT85ZH60EPcdEYZZrUKNVrpm/HE
vZ7Q1ThXKswh0uoXpFdHmlLuF38gVDJ/S94og83IulvNvXACOfahZaap6yyZPrPT
/zKnMmPc
-----END CERTIFICATE REQUEST-----
```

```
#如果不一致，在 CA 为申请用户 server 签发证书时，会出现以下提示
```

```
D:\Pros\OpenSSL-Win64\bin>openssl ca -in demoCA\users\server.csr -out demoCA\users\server.pem.crt -cert demoCA\ca\ca.pem.crt -keyfile demoCA\ca\ca.key -config openssl.cfg
Using configuration from openssl.cfg
Enter pass phrase for ca.key:
Check that the request matches the signature
Signature ok
The organizationName field is different between
CA certificate (Test) and the request (TEST)
```

```
#策略配置方式，修改 openssl.cfg 文件，这个地方
```



```
# For the CA policy
[ policy_match ]
countryName		= match
stateOrProvinceName	= match
organizationName	= match
organizationalUnitName	= optional
commonName		= supplied
emailAddress		= optional
```

```
#demoCA\sign.txt 的内容如下
```

指定单一域名：

```
  subjectAltName = DNS.1:test.server.cn
```

如果要通配：

```
  subjectAltName = DNS.1:server.cn,DNS.2:*.server.cn
```

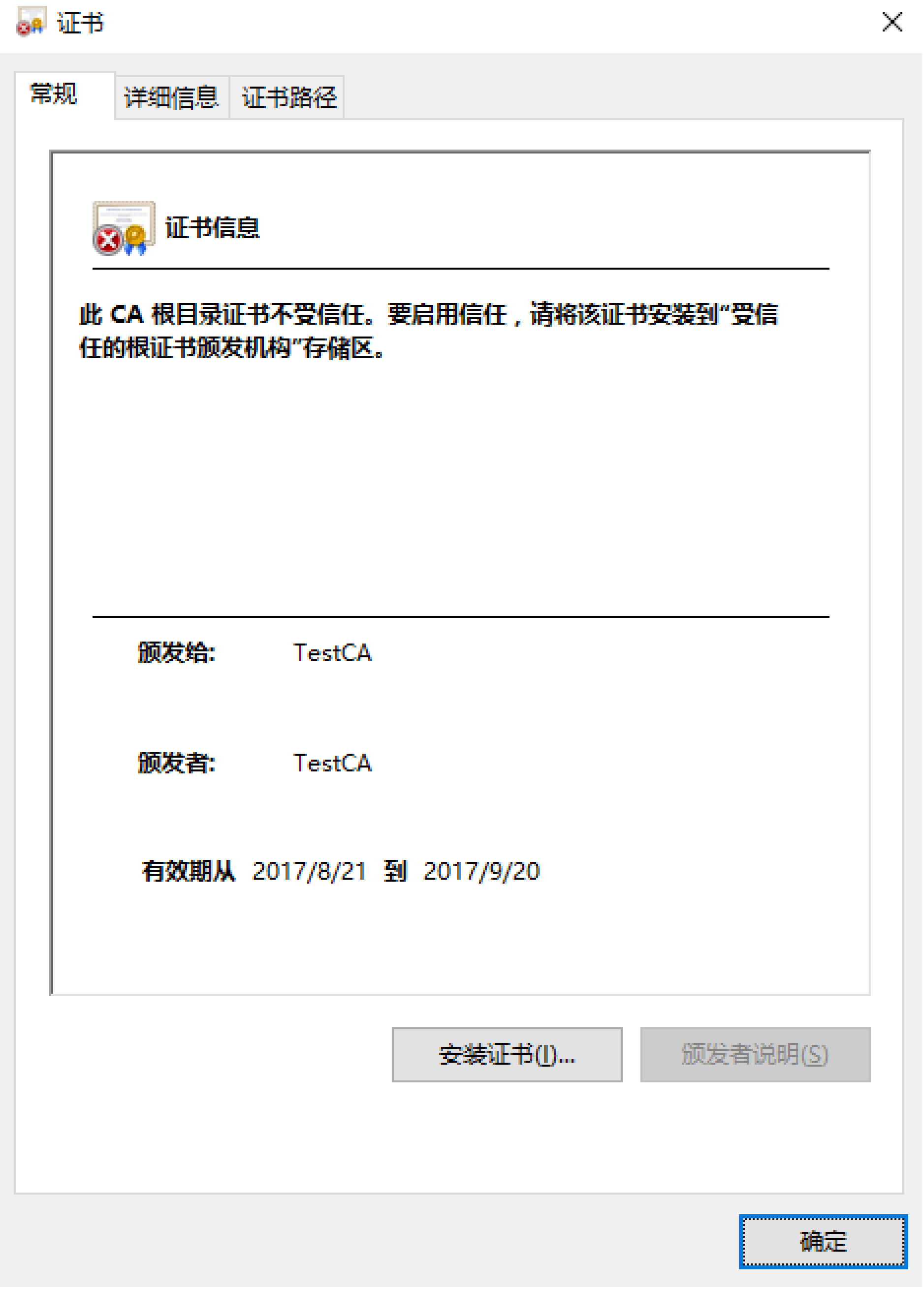
```
#一致的情况下，CA 为申请用户 server 签发证书成功提示如下
```

```
D:\Pros\OpenSSL-Win64\bin>openssl ca -in demoCA\users\server.csr -out demoCA\us
ers\server.pem.crt -extfile demoCA\sign.txt -days 365 -cert demoCA\ca\ca.pem.cr
t -keyfile demoCA\ca\ca.key -config openssl.cfg

Using configuration from openssl.cfg

Enter pass phrase for ca.key:

Check that the request matches the signature

Signature ok

Certificate Details:

        Serial Number: 1 (0x1)

        Validity

            Not Before: Aug 21 11:24:44 2017 GMT

            Not After : Aug 21 11:24:44 2018 GMT

        Subject:

            countryName               = CN

            stateOrProvinceName        = Beijing

            organizationName           = TEST

            organizationalUnitName     = UnitName

            commonName                = www.server.com

            emailAddress              = test@server.com

        X509v3 extensions:

            X509v3 Basic Constraints:

                CA:FALSE

            Netscape Comment:

                OpenSSL Generated Certificate

            X509v3 Subject Key Identifier:

                25:18:C7:31:80:52:E1:CF:F7:05:1D:A6:54:8D:F3:FF:C6:93:6E:98

            X509v3 Authority Key Identifier:

                keyid:86:CA:B6:14:24:B7:93:18:48:70:FE:7A:1C:94:8F:DA:B3:F9:49:8
8


Certificate is to be certified until Aug 21 11:24:44 2018 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

#生成后，查看users 目录，多了一个证书server.pem.crt
#同时在newcerts 里有一个01.pem.crt，是server.pem.crt 的备份
```

3.3 windows 下查看证书 crt 文件

3.3.1 直接查看根证书和 server 的证书，是这样的，提示不受信任



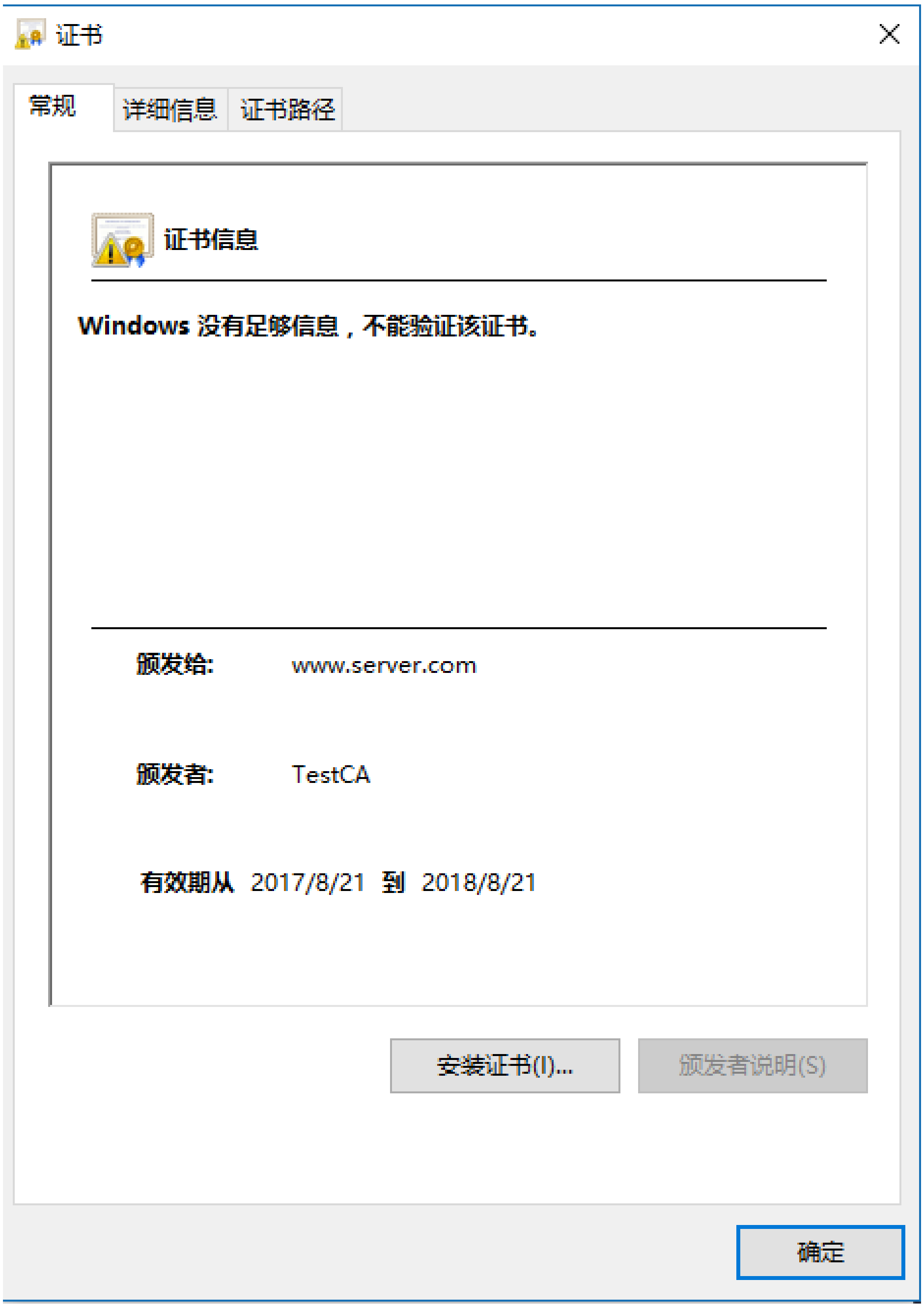
证书信息

此 CA 根目录证书不受信任。要启用信任，请将该证书安装到"受信任的根证书颁发机构"存储区。

颁发给: TestCA

颁发者: TestCA

有效期从 2017/8/21 **到** 2017/9/20



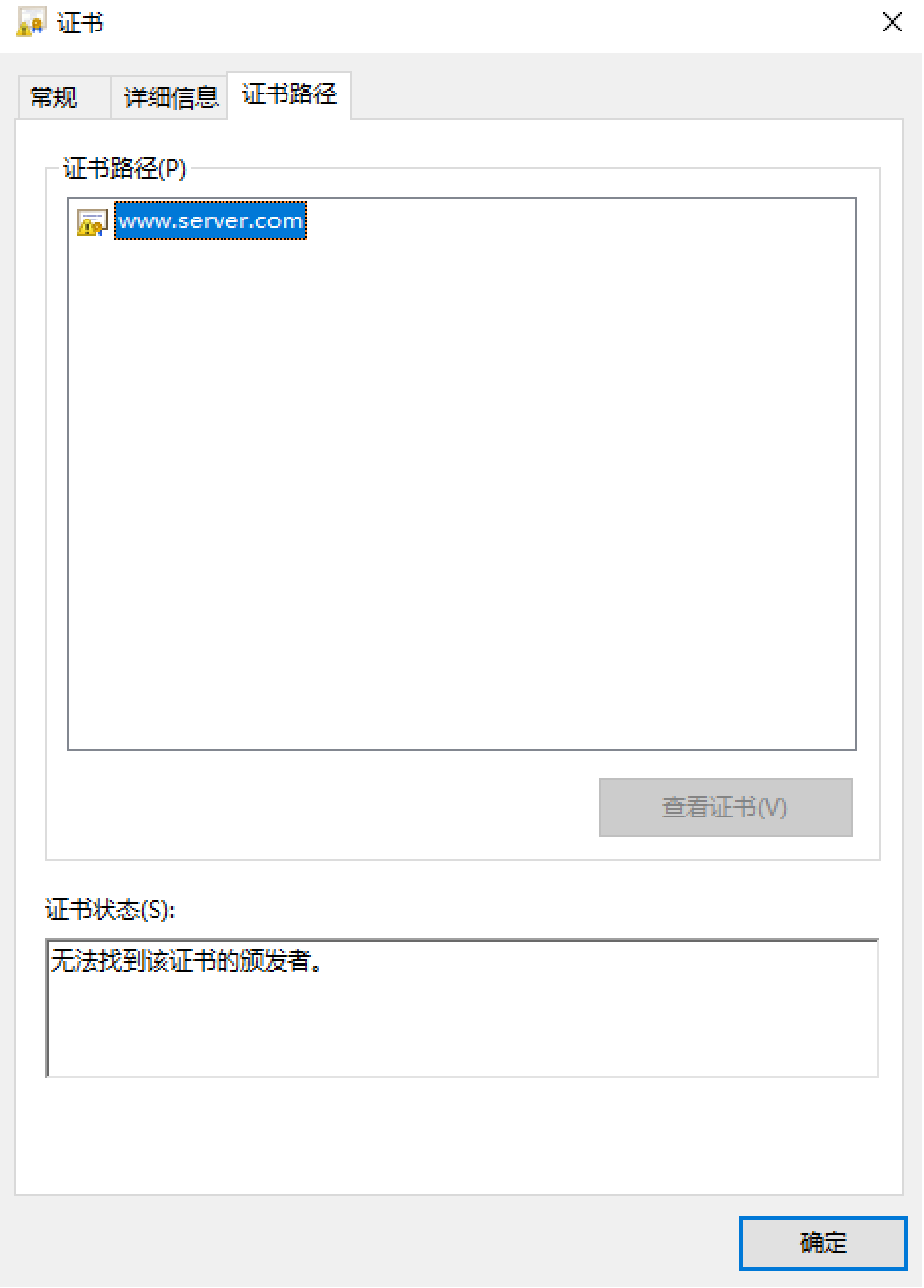
证书信息

Windows 没有足够信息，不能验证该证书。

颁发给: www.server.com

颁发者: TestCA

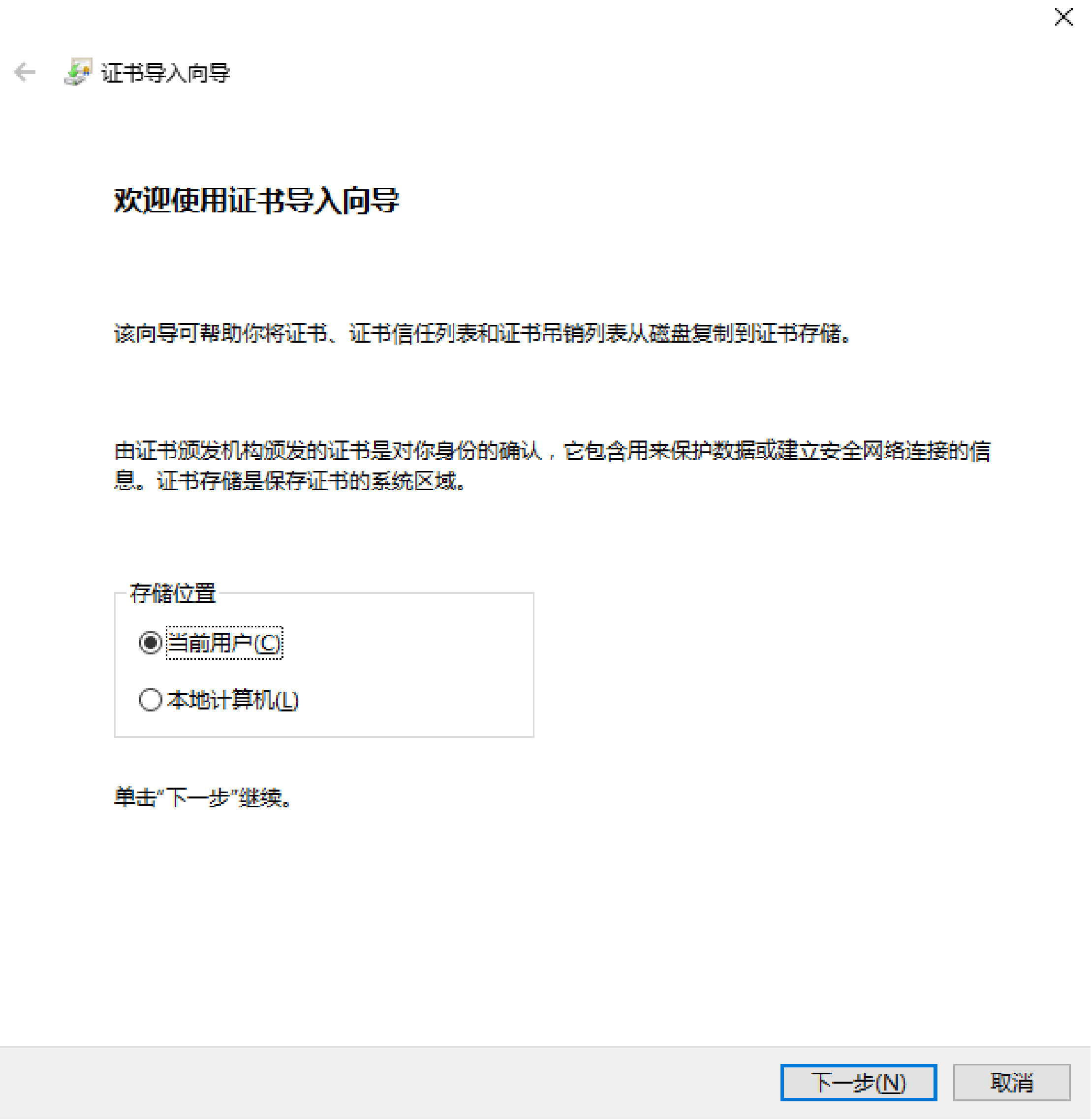
有效期从 2017/8/21 到 2018/8/21



证书状态(S):

无法找到该证书的颁发者。

3.3.2 导入根证书的步骤



证书导入向导

欢迎使用证书导入向导

该向导可帮助你将证书、证书信任列表和证书吊销列表从磁盘复制到证书存储。

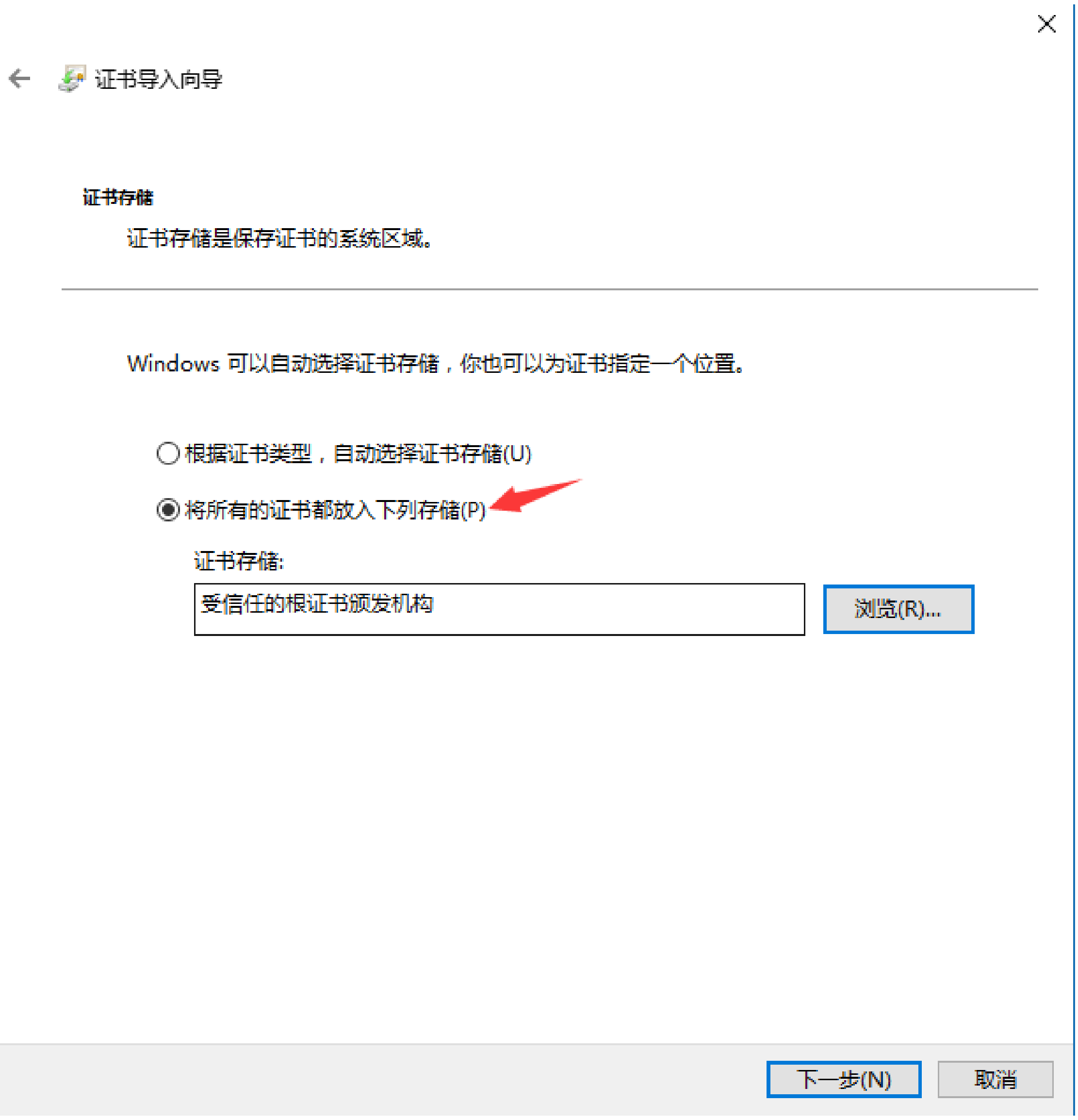
由证书颁发机构颁发的证书是对你身份的确认，它包含用来保护数据或建立安全网络连接的信息。证书存储是保存证书的系统区域。

存储位置

● 当前用户(C)

○ 本地计算机(L)

单击“下一步”继续。



证书导入向导

证书存储

证书存储是保存证书的系统区域。

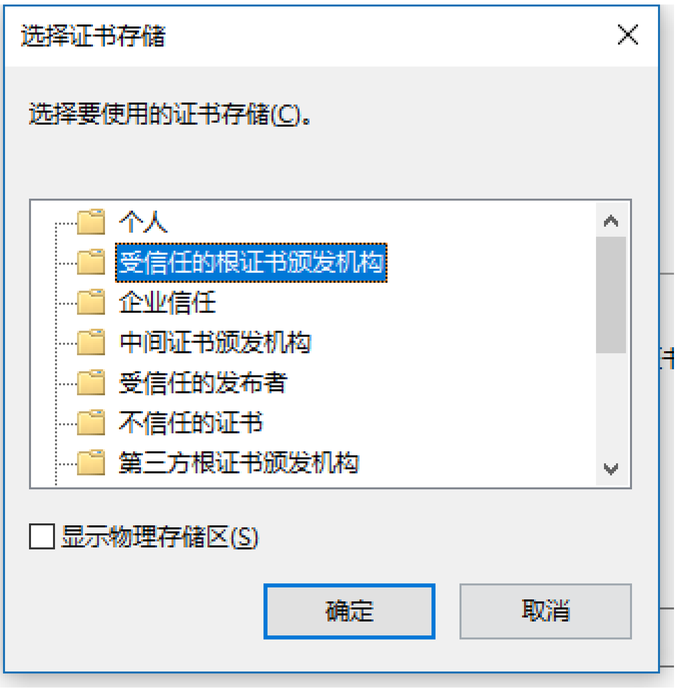
Windows 可以自动选择证书存储，你也可以为证书指定一个位置。

○ 根据证书类型，自动选择证书存储(U)

● 将所有的证书都放入下列存储(P)

证书存储:

受信任的根证书颁发机构

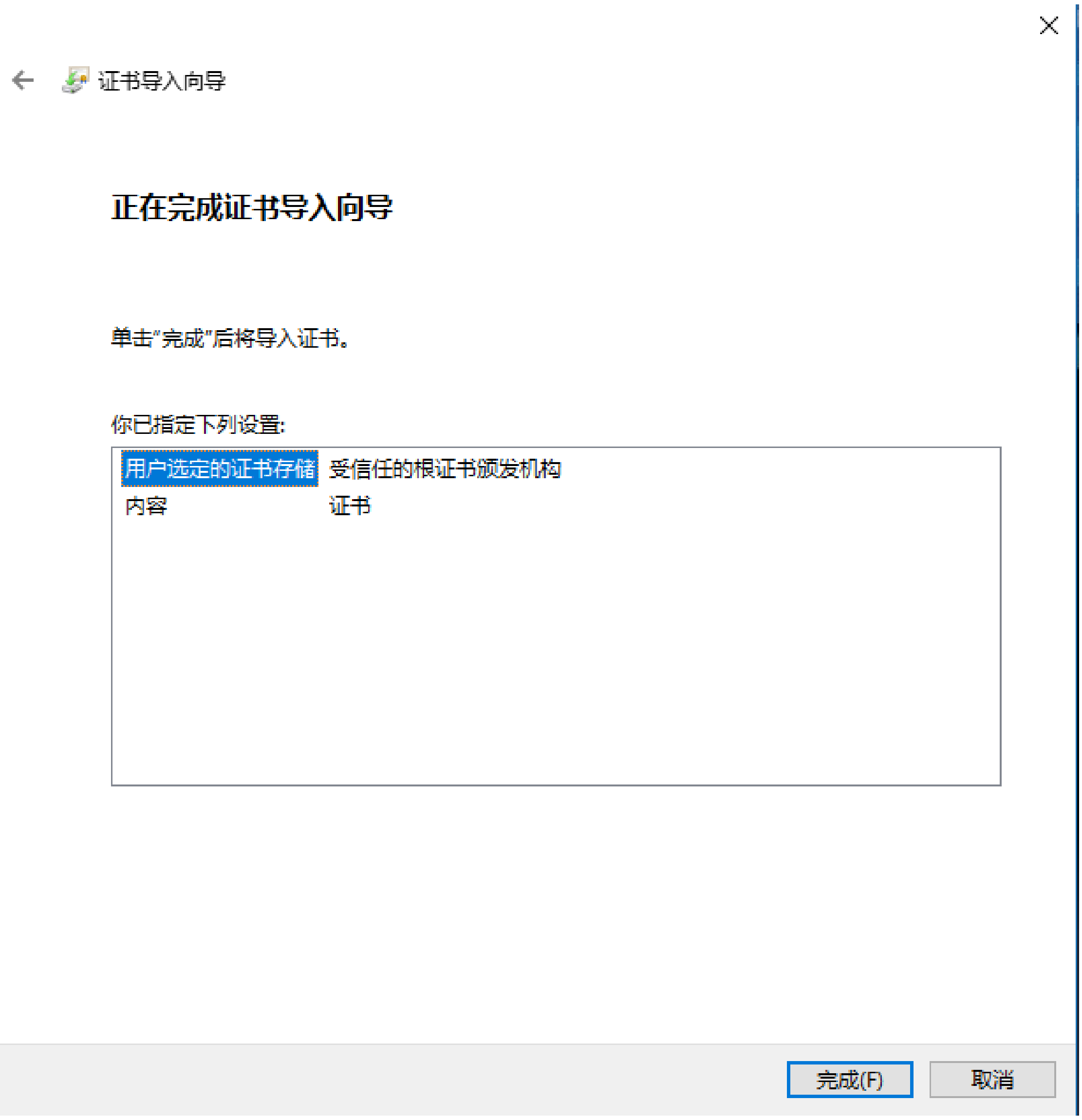


选择证书存储

选择要使用的证书存储(C)。

- 个人
- 受信任的根证书颁发机构
- 企业信任
- 中间证书颁发机构
- 受信任的发布者
- 不信任的证书
- 第三方根证书颁发机构

□ 显示物理存储区(S)

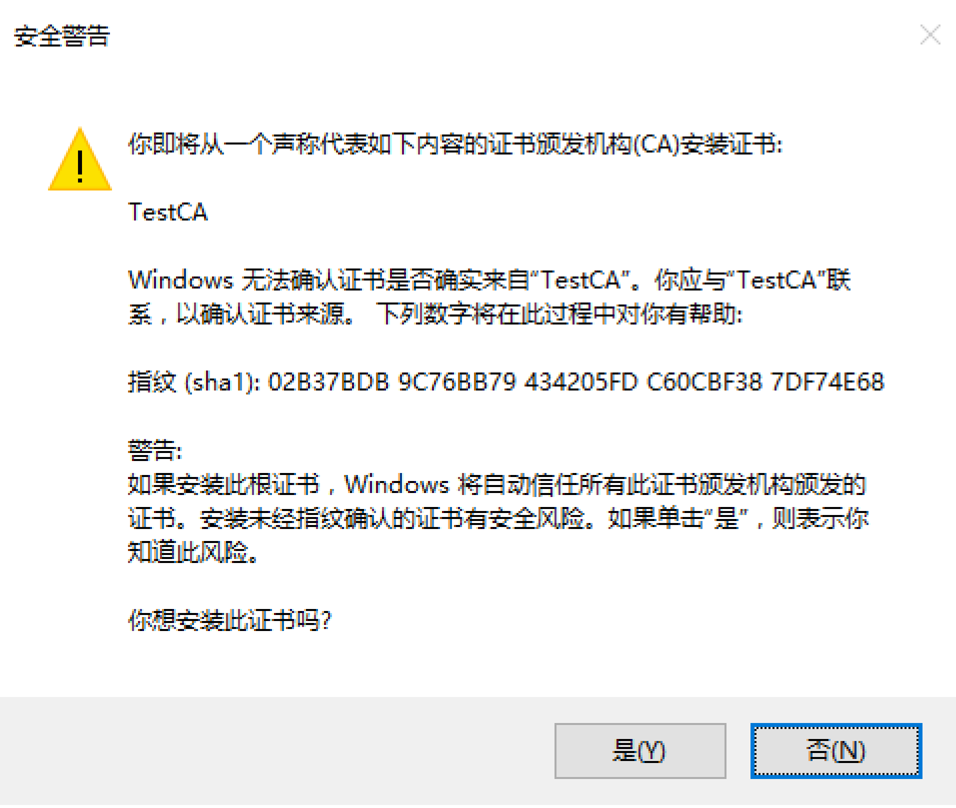


正在完成证书导入向导

单击"完成"后将导入证书。

你已指定下列设置:

用户选定的证书存储	受信任的根证书颁发机构
内容	证书



你即将从一个声称代表如下内容的证书颁发机构(CA)安装证书:

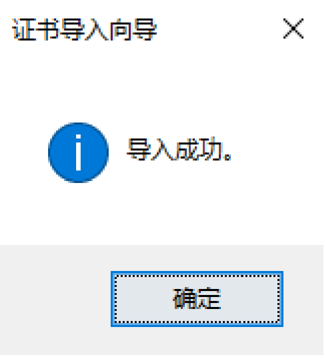
TestCA

Windows 无法确认证书是否确实来自"TestCA"。你应与"TestCA"联系，以确认证书来源。 下列数字将在此过程中对你有帮助:

指纹 (sha1): 02B37BDB 9C76BB79 434205FD C60CBF38 7DF74E68

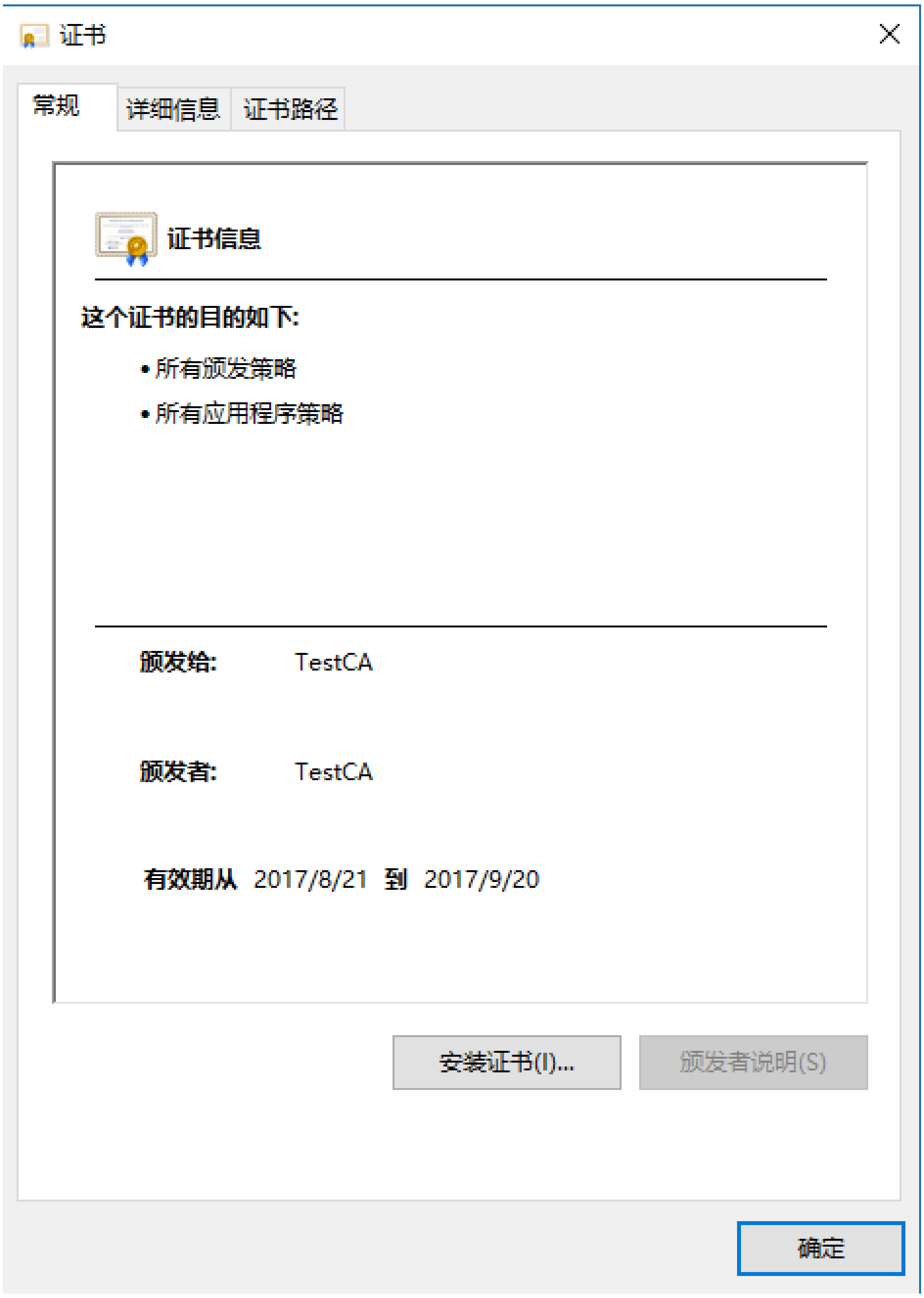
警告:
如果安装此根证书，Windows 将自动信任所有此证书颁发机构颁发的证书。安装未经指纹确认的证书有安全风险。如果单击"是"，则表示你知道此风险。

你想安装此证书吗?



导入成功。

3.3.3 导入成功后，重新查看



证书信息

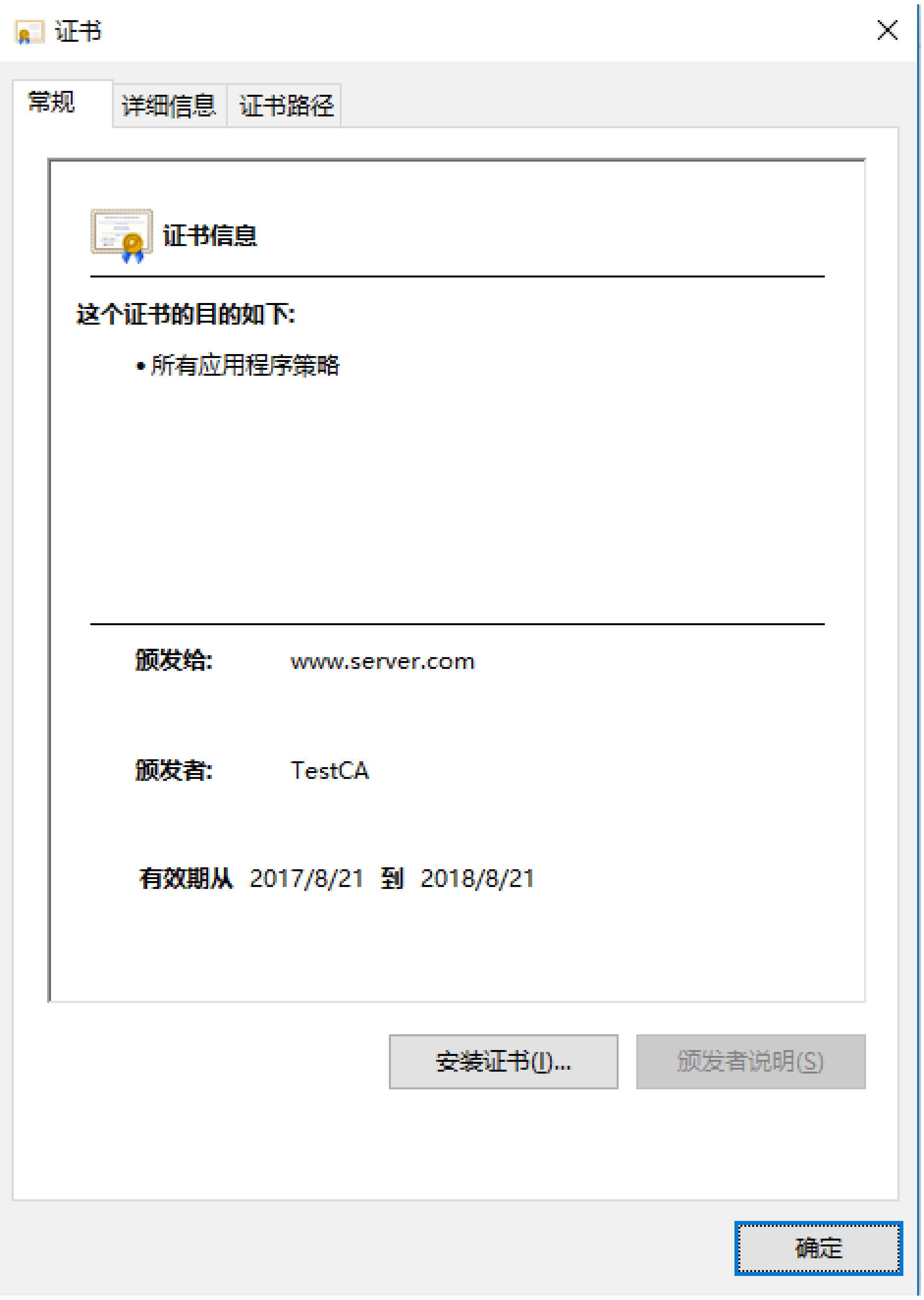
这个证书的目的如下:

- 所有颁发策略
- 所有应用程序策略

颁发给: TestCA

颁发者: TestCA

有效期从 2017/8/21 **到** 2017/9/20



证书

证书信息

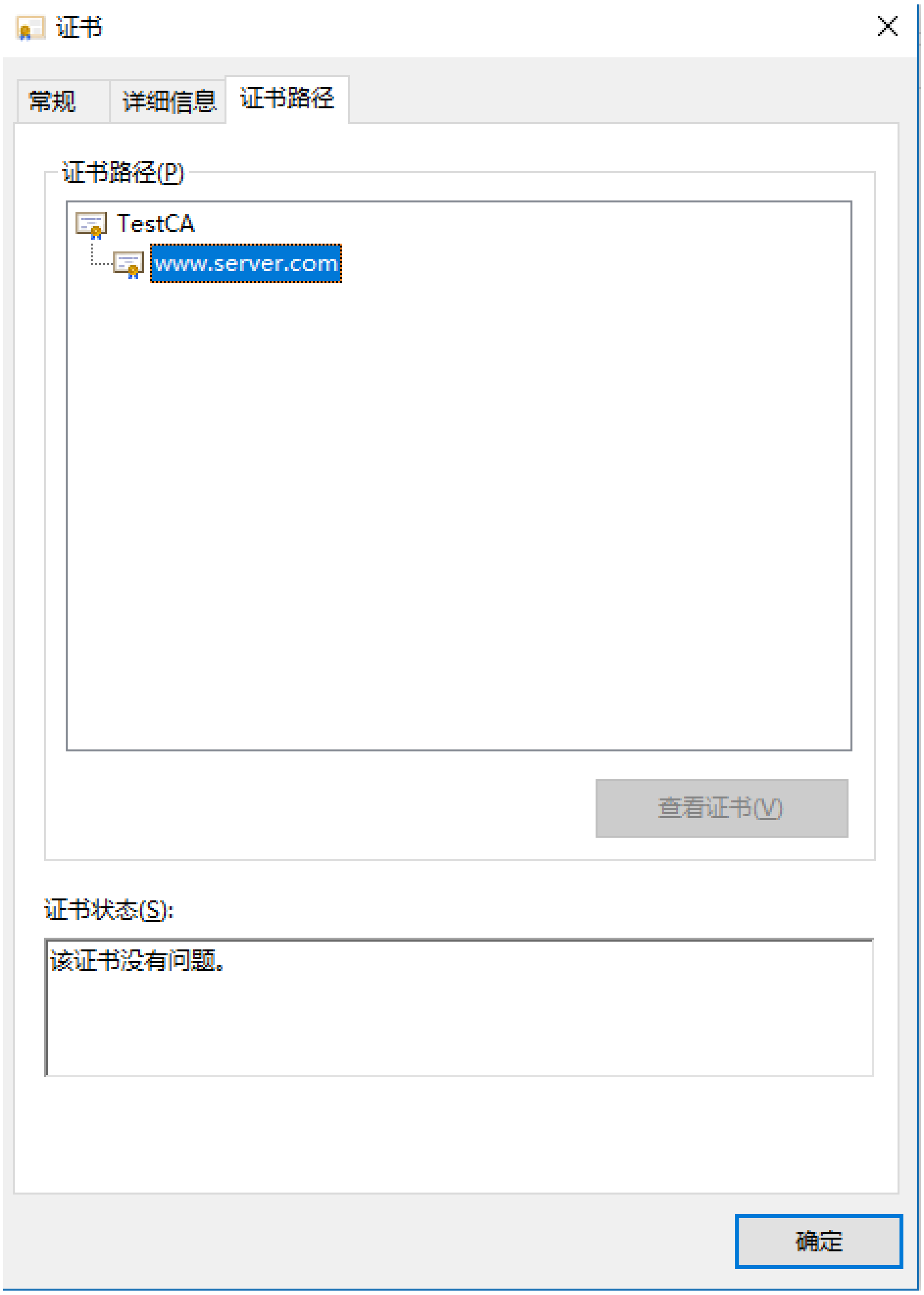
这个证书的目的如下:

- 所有应用程序策略

颁发给: www.server.com

颁发者: TestCA

有效期从 2017/8/21 到 2018/8/21



该证书没有问题。

—TODO—

证书格式转换

签名

验签

加密

解密

对称加解密文件

生成 HASH 值