**中间件（C语言）签证书和CRL的使用说明**

本文对使用中间件（C语言）的签证书和签CRL等函数进行使用说明。JAVA版本的使用的是另外的方式去实现的，不在本文档中描述。

# 1 签证书

## 1.1 NetcaPKICertSignCert

原型：

int NETCADLLAPI NetcaPKICertSignCert(NETCA\_PKI\_CERTIFICATE\_HANDLE issuerCert,const char \*templateInfo,const char \*dataInfo,void \*context,NetcaSignCallback\_Ptr signCallback,NETCA\_PKI\_CERTIFICATE\_HANDLE \*pCert)

参数：

issuerCert：颁发者证书，应该带有密钥对。如果是自签证书，则为NULL，这时需要使用signCallback进行签名

templateInfo：输入参数，模板信息，UTF-8编码的JSON格式的对象

dataInfo：输入参数，相关的数据信息，UTF-8编码的JSON格式的对象

context：signCallback函数使用的上下文信息

signCallback：回调的签名函数，如果不为NULL，则使用该回调函数进行签名，否则使用issuerCert关联的私钥进行签名。

pCert：输出参数，成功返回签发出的证书

返回值：

成功返回NETCA\_PKI\_SUCCESS，如果是templateInfo有问题返回NETCA\_PKI\_TEMPLATE\_FAIL，dataInfo有问题返回NETCA\_PKI\_TEMPLATE\_DATA\_FAIL。出错返回其他值。

说明：

根据模板来签证书。模板使用JSON来表示，详见1.2。把模板分为templateInfo和dataInfo，主要是考虑到让templateInfo可以重用。

通常自签证书，issuerCert为NULL，signCallback设置为真正的签名函数。签其他证书，issuerCert不为NULL，且带有密钥对，signCallback为NULL。

本函数没有对模板进行严格检查，可能会签发出一些不符合标准的证书，因此保证模板的正确性就比较关键。

NetcaSignCallback\_Ptr的定义如下：

typedef int (NETCADLLAPI\* NetcaSignCallback\_Ptr)(void \*context,int algo,const void \*param,const unsigned char \*data,int dataLen,unsigned char \*\*signature,int \*signatureLen);

参数：

context：上下文信息

algo：输入参数，签名算法

param：输入参数，签名算法参数

data：输入参数，要签名的数据

dataLen：输入参数，data的字节数

signature：输出参数，成功返回的签名值。其必须能使用NetcaPKICryptoFreeMemory来释放

signatureLen：输出参数，成功返回签名值的字节数

返回值：

成功返回NETCA\_PKI\_SUCCESS，失败返回其他值

说明：

进行签名

## 1.2 模板说明

templateInfo和dataInfo都是JSON对象的编码，而且采用UTF-8来编码的。它们是相关联的，缺一不可。下文对templateInfo使用证书模板来称呼，dataInfo使用数据模板来称呼。

### 1.2.1版本号

版本号只需要证书模板，不需要数据模板。

版本号的证书模板为version，对应的类型为字符串。如果不存在，则表示使用版本3。其值如下表所示：

|  |  |
| --- | --- |
| 值 | 说明 |
| V1 | 版本1 |
| V2 | 版本2 |
| V3 | 版本3 |

由于当前基本上都是使用版本3的证书，因此，直接省略version这一项即可。

### 1.2.2证书序列号

证书序列号的证书模板为serialNumber，对应的类型为对象。

当前支持三种方式进行设置证书序列号:

1. 随机产生的非负序列号。

这种情况下，只需要证书模板，不需要数据模板。

证书模板取值如下：

|  |  |  |
| --- | --- | --- |
| 名称 | 类型 | 说明 |
| type | 字符串 | 值必须为random |
| value | 数字(整数) | 表示产生的随机数的长度，编码后可能长度会小于该长度 |

例如：

"serialNumber":{

"type":"random",

"value":16

}

表示产生16个字节的随机数作为序列号。

1. 使用固定的序列号

这种情况下，只需要证书模板，不需要数据模板。

证书模板取值如下：

|  |  |  |
| --- | --- | --- |
| 名称 | 类型 | 说明 |
| type | 字符串 | 值必须为hex |
| external | 布尔值 | 值必须为false |
| value | 字符串 | Hex编码的序列号 |

注意：很少需要使用这样的模板，这种方式可以指定负的序列号

例如：

"serialNumber":{

"type":"hex",

"external":false,

"value":"12345678"

}

表示设置序列号为"\x12\x34\x56\x78"。

1. 从数据模板中得到序列号

证书模板取值如下：

|  |  |  |
| --- | --- | --- |
| 名称 | 类型 | 说明 |
| type | 字符串 | 值必须为hex |
| external | 布尔值 | 值必须为true |

数据模板为serialNumber，对应的类型为字符串，其值为Hex编码的序列号，这种方式可以使用负的序列号。这样就可以自己先生成唯一的序列号，在数据模板中，设置好值，用相同的证书模板来签证书。

例如：

证书模板，如下：

"serialNumber":{

"type":"hex",

"external":true

}

对应的数据模板，如下：

"serialNumber":"12345678"

表示设置序列号为"\x12\x34\x56\x78"。

### 1.2.3 签名算法

签名算法只需要证书模板，不需要数据模板。

签名算法的证书模板为signatureAlgorithm，对应的类型为对象。

证书模板取值如下：

|  |  |  |
| --- | --- | --- |
| 名称 | 类型 | 说明 |
| algo | 字符串 | 签名算法的名称或者OID |
| param | 字符串 | 如果不存在表示没有算法参数，如果为NULL，则表示算法参数为NULL，否则为算法参数编码再进行Hex编码 |

例如：

"signatureAlgorithm":{

"algo":"SHA256WithRSAEncryption",

"param":"NULL"

}

表示SHA256WithRSAEncryption

"signatureAlgorithm":{

"algo":"ECDSAWithSHA256"

}

表示ECDSAWithSHA256

### 1.2.4颁发者名

对于自签证书，颁发者名和主体名一样

对于其他证书，颁发者名为颁发者证书中的主体名

证书模板的selfsign项用来控制是否自签证书，其类型为布尔类型。如果不存在，则不是自签证书。

例如：

"selfsign":true

表示自签证书

### 1.2.5 有效期

证书模板有三个项和有效期相关: inIssuerValidity、notBefore和notAfter。

inIssuerValidity用来限制是否签发出的证书的有效期必须在颁发者证书的有效期之内。其类型为布尔类型。如果不存在，则表示签发出的证书的有效期必须在颁发者证书的有效期之内。

例如：

"inIssuerValidity":false

表示证书的有效期可以超出颁发者证书的有效期

notBefore表示有效期开始时间，是个JSON对象，有三种方式来设置：

1. 使用当前时间

不使用notBefore项。或者证书模板的取值如下：

|  |  |  |
| --- | --- | --- |
| 名称 | 类型 | 说明 |
| type | 字符串 | 值必须为now |

1. 使用固定的开始时间

证书模板的取值如下：

|  |  |  |
| --- | --- | --- |
| 名称 | 类型 | 说明 |
| type | 字符串 | 值必须为time |
| external | 布尔值 | 不存在，或者值必须为false |
| value | 字符串 | 时间，格式为YYYYMMDDhhmmssZ |

例如：

"notBefore":{

"type":"time",

"value":"20180501020314Z"

}

表示设置有效期开始时间为UTC时间:2018年5月1日2点3分14秒。

1. 从数据模板中获取开始时间

证书模板的取值如下：

|  |  |  |
| --- | --- | --- |
| 名称 | 类型 | 说明 |
| type | 字符串 | 值必须为time |
| external | 布尔值 | 值必须为true |

数据模板为notBefore，对应的类型为字符串，表示时间，格式为YYYYMMDDhhmmssZ。

例如：

证书模板：

"notBefore":{

"type":"time",

"external":true

}

数据模板

"notBefore":"20180502112345Z"

表示设置有效期开始时间为UTC时间:2018年5月2日11点23分45秒。

notAfter表示有效期结束时间，是个JSON对象，有三种方式来设置：

1. 设置有效期结束时间为有效期开始时间加上某个值。比如说，有效期为一年，也就是有效期结束时间为有效期开始时间加上1年。

证书模板的取值如下：

|  |  |  |
| --- | --- | --- |
| 名称 | 类型 | 说明 |
| type | 字符串 | 值必须为interval |
| value | 对象 | 有效期的时间 |

value的取值，如下表所示：

|  |  |  |
| --- | --- | --- |
| 名称 | 类型 | 说明 |
| year | 数字(整数) | 年，可选 |
| month | 数字(整数) | 月，可选 |
| day | 数字(整数) | 日，可选 |
| hour | 数字(整数) | 小时，可选 |

例如：

"notAfter":{

"type":"interval",

"value":{

"year":1

}

}

表示有效期为1年

1. 设置有效期结束时间为某个固定的时间

证书模板的取值如下：

|  |  |  |
| --- | --- | --- |
| 名称 | 类型 | 说明 |
| type | 字符串 | 值必须为time |
| external | 布尔值 | 不存在，或者值必须为false |
| value | 字符串 | 时间，格式为YYYYMMDDhhmmssZ |

例如：

"notAfter":{

"type":"time",

"value":"99991231235959Z"

}

表示不会过期

1. 从数据模板中获取有效期结束时间

证书模板的取值如下：

|  |  |  |
| --- | --- | --- |
| 名称 | 类型 | 说明 |
| type | 字符串 | 值必须为time |
| external | 布尔值 | 值必须为true |

数据模板为notAfter，对应的类型为字符串，表示时间，格式为YYYYMMDDhhmmssZ。

例如：

证书模板：

"notAfter":{

"type":"time",

"external":true

}

数据模板

"notAfter":"20480502112345Z"

表示设置有效期结束时间为UTC时间:2018年5月2日11点23分45秒。

### 1.2.6主体名

主体名的证书模板为subject，对应的类型为对象。

主体名的设置有两种方式：

1. 从数据模板中获取整个主体名的编码

证书模板的取值如下：

|  |  |  |
| --- | --- | --- |
| 名称 | 类型 | 说明 |
| type | 字符串 | 值必须为whole |

数据模板为subject，对应的类型为字符串，表示Hex格式的主体名的编码

例如：

证书模板

"subject":{

"type":"whole"

}

数据模板

"subject":"3000"

表示空的主体名

1. 证书模板限定主体名中属性的项及其编码，由数据模板中对应数据填充其值。

证书模板的取值如下：

|  |  |  |
| --- | --- | --- |
| 名称 | 类型 | 说明 |
| type | 字符串 | 值必须为default |
| value | 对象数组 | 每一项为一个属性项 |

value的属性项的取值如下：

|  |  |  |
| --- | --- | --- |
| 名称 | 类型 | 说明 |
| option | 字符串 | 如果为fixed或者不存在，表示该属性为固定值，其值在证书模板中。如果为optional，表示该属性是可选的，其值在数据模板中。如果为required则表示该属性是必需的其值在数据模板中。 |
| attrType | 字符串 | 属性的名称（C、CN等）或者属性的OID |
| encodeType | 字符串 | 属性值的编码类型，PrintableString、IA5String、UTF8String等。如果是PrintableString->UTF8String则表示如果属性值可以用PrintableString表示则用PrintableString编码，不能则用UTF8String编码 |
| multiValue | 布尔类型 | 如果为true则表示可以使用多个值，如果为false则表示仅能使用一个值。不存在表示false |
| attrValue | 字符串或者字符串数组 | 可选。仅在option为fixed时有效。表示属性值 |

数据模板的为subject，对应的类型为对象。该对象中，以OID为名称，得到的值为对应属性的值。属性的值是字符串或者字符串数组。比如：属性CN的OID为"2.5.4.3"，其值这样设置：

"subject":{

"2.5.4.3":"具体CN的值",

……

}

无论属性值在证书模板或者数据模板中，属性值是这样设置的。如果是单值属性，则为字符串，表示该属性值。如果是多值属性，如果属性值为字符串，则表示该属性只有一个值，如果属性值为字符串数组，则每个字符串数组的值为一个属性值。

这种方式不能处理一个RDN有多个属性的情况。对于存在多个(>1)属性值,会对每个属性值产生一个RDN。

例如：

证书模板

"subject":{

"type":"default",

"value":[

{

"option":"fixed",

"attrType":"C",

"encodeType":"PrintableString",

"attrValue":"CN"

},

{

"option":"optional",

"attrType":"O",

"encodeType":"UTF8String"

},

{

"option":"required",

"attrType":"CN",

"encodeType":"UTF8String",

"multiValue":true

}

]

}

表示主体名中，C项的值固定为CN，编码为PrintableString。O项是可选的，编码为UTF8String，其值从数据模板中读取。CN项是必须的编码为UTF8String，其值从数据模板中读取，并且可以存在多个CN。

对应的数据模板如果为：

"subject":{

"2.5.4.3":["用户证书","test"],

"2.5.4.10":"用户组织"

}

其中，2.5.4.3为CN的OID, 2.5.4.10为O的OID。

表示O的值为用户组织，CN的值有两个，一个为用户证书，一个为test。

最终的编码如下:

SEQUENCE {

SET {

SEQUENCE {

OBJECT IDENTIFIER countryName (2 5 4 6)

PrintableString 'CN'

}

}

SET {

SEQUENCE {

OBJECT IDENTIFIER organizationName (2 5 4 10)

UTF8String '用户组织'

}

}

SET {

SEQUENCE {

OBJECT IDENTIFIER commonName (2 5 4 3)

UTF8String '用户证书'

}

}

SET {

SEQUENCE {

OBJECT IDENTIFIER commonName (2 5 4 3)

UTF8String 'test'

}

}

}

这是dumpasn1显示的结果已经取得前面的位移和长度。

### 1.2.7 公钥

公钥在证书模板中没有相应的项，其值总是从数据模板中获取。

数据模板中，公钥为subjectPublicKeyInfo项，字符串类型，其值为SubjectPublicKeyInfo的DER编码后再进行Hex编码

### 1.2.8 issuerUniqueID

不支持。这是没人用的项

### 1.2.9 subjectUniqueID

不支持。这是没人用的项

### 1.2.10 证书扩展

证书模板中，证书扩展是extensions项，这是一个对象数组，数组中每一项表示一个证书扩展。

证书模板中，每个扩展有以下公用的项

|  |  |  |
| --- | --- | --- |
| 名称 | 类型 | 说明 |
| name | 字符串 | 扩展的名称或者扩展的OID |
| critical | 布尔类型 | 为真表示关键扩展，为假表示非关键扩展。不存在表示非关键扩展 |

数据模板中，证书扩展是extensions项，这是一个对象，其中，每个扩展OID对应的项表示对应扩展需要的值。比如：

"extensions":{

"2.5.29.14":"HEX编码的主体密钥标识",

……

}

其中2.5.29.14为主体密钥标识的OID。

下面逐个说明各个扩展的模板

#### 1.2.10.1 颁发者密钥标识符

颁发者密钥标识符只有证书模板，没有数据模板

证书模板的取值如下：

|  |  |  |
| --- | --- | --- |
| 名称 | 类型 | 说明 |
| name | 字符串 | 值必须为Authority Key Identifier |
| encodeType | 字符串 | 值必须为default |
| value | 字符串 | 如果为keyid则表示仅使用密钥标识符，如果为issuer\_issuer\_sn则表示仅使用颁发者的颁发者名和序列号。如果为keyid,issuer\_issuer\_sn则是所有的三项都使用 |

注意：对于自签证书来说，颁发者密钥标识符如果选用密钥标识符则必须放到主体密钥标识符扩展之后，否则会签证书失败。

例如：

{

"name":"Authority Key Identifier",

"encodeType":"default",

"value":"keyid"

}

表示颁发者密钥标识符仅使用密钥标识符。

#### 1.2.10.2 主体密钥标识符

主体密钥标识符有两种方式：

1. 根据公钥来产生

证书模板的取值如下：

|  |  |  |
| --- | --- | --- |
| 名称 | 类型 | 说明 |
| name | 字符串 | 值必须为Subject Key Identifier |
| encodeType | 字符串 | 值必须为default |
| value | 字符串 | hash:Hash算法名:低位数:二进制前缀。低位数和二进制前缀是可选的 |

例如：

{

"name":"Subject Key Identifier",

"encodeType":"default",

"value":"hash:SHA-1"

}

表示主体密钥标识符使用RFC5280的方法1。

{

"name":"Subject Key Identifier",

"encodeType":"default",

"value":"hash:SHA-1:61:0100"

}

表示主体密钥标识符使用RFC5280的方法2。

1. 从数据模板中来获取

证书模板的取值如下：

|  |  |  |
| --- | --- | --- |
| 名称 | 类型 | 说明 |
| name | 字符串 | 值必须为Subject Key Identifier |
| encodeType | 字符串 | 值必须为keyid |
| option | 字符串 | 如果为optional表示该扩展可选，其值由数据模板来获取。如果为required，则表示该扩展是必需的，其值由数据模板来获取。 |

对应的数据模板的取值为密钥标识符的Hex编码的字符串。

例如：

证书模板：

{

"name":"Subject Key Identifier",

"option":"required",

"encodeType":"keyid"

}

对应的数据模板：

"2.5.29.14":"12abcdef"

则主体密钥标识符为"\x12\xab\xcd\xef"

#### 1.2.10.3 密钥用法

密钥用法只有证书模板，没有数据模板

密钥用法有两种方式：

1. 使用名称

证书模板的取值如下：

|  |  |  |
| --- | --- | --- |
| 名称 | 类型 | 说明 |
| name | 字符串 | 值必须为Key Usage |
| encodeType | 字符串 | 值必须为name |
| value | 字符串数组 | 数组中每一项表示一个密钥用法的名称 |

例如：

{

"name":"Key Usage",

"critical":true,

"encodeType":"name",

"value":[

"digitalSignature","contentCommitment"

]

}

表示关键的密钥用法扩展，其中第0和第1位置位

1. 按照位逐位进行设置

证书模板的取值如下：

|  |  |  |
| --- | --- | --- |
| 名称 | 类型 | 说明 |
| name | 字符串 | 值必须为Key Usage |
| encodeType | 字符串 | 值必须为bit |
| value | 布尔数组 | 数组中每一项表示一位，其值为真，则该位置位 |

例如：

{

"name":"Key Usage",

"critical":true,

"encodeType":"bit",

"value":[

true,true

]

}

表示关键的密钥用法扩展，其中第0和第1位置位

#### 1.2.10.4 证书策略

证书策略只有证书模板，没有数据模板

证书模板的取值如下：

|  |  |  |
| --- | --- | --- |
| 名称 | 类型 | 说明 |
| name | 字符串 | 值必须为Certificate Policies |
| value | 对象数组 | 数组中每一项表示一个证书策略 |

value的证书策略项的取值如下:

|  |  |  |
| --- | --- | --- |
| 名称 | 类型 | 说明 |
| policyIdentifier | 字符串 | 证书策略的OID |
| policyQualifiers | 对象数组 | 数组中每一项表示一个证书策略限定符。可选 |

证书策略限定符的取值如下:

|  |  |  |
| --- | --- | --- |
| 名称 | 类型 | 说明 |
| policyQualifierId | 字符串 | cPSuri、userNotice或者证书策略限定符的OID |
| qualifier | 根据policyQualifierId的值来确定 | 如果是cPSuri，则为字符串，表示CPS的URL。如果是userNotice则是对象。否则是字符串，表示qualifier的DER编码的Hex编码 |

userNotice的qualifier的取值如下:

|  |  |  |
| --- | --- | --- |
| 名称 | 类型 | 说明 |
| explicitText | 对象 | 否认书，可选 |
| noticeRef | 对象 | 引用，可选 |

explicitText的取值如下:

|  |  |  |
| --- | --- | --- |
| 名称 | 类型 | 说明 |
| encodeType | 字符串 | 编码类型。UTF8String等 |
| value | 字符串 | 实际的值 |

noticeRef的取值如下:

|  |  |  |
| --- | --- | --- |
| 名称 | 类型 | 说明 |
| organization | 对象 | 和explicitText的类型一样 |
| noticeNumbers | 整数数组 | 引用号 |

例如：

{

"name":"Certificate Policies",

"value":[

{

"policyIdentifier":"1.2.3.4.5.6.7.8.9",

"policyQualifiers":[

{

"policyQualifierId":"cPSuri", "qualifier":"http://www.test.com/testcps"

}

]

}

]

}

表示有一个证书策略，其策略OID为1.2.3.4.5.6.7.8.9，对应的CPS的URI为http://www.test.com/testcps

#### 1.2.10.5 策略映射

策略映射只有证书模板，没有数据模板

证书模板的取值如下：

|  |  |  |
| --- | --- | --- |
| 名称 | 类型 | 说明 |
| name | 字符串 | 值必须为Policy Mappings |
| value | 两项的字符串数组的数组 | 数组中的每一项是个映射关系，其中第0项的值为颁发者域的策略OID，第1项的值为主体域的策略OID |

例如：

{

"name":"Policy Mappings",

"critical":true,

"value":[

["1.2.3.4","1.2.3.5"]

]

}

表示关键的策略映射扩展，其中包含一项策略映射，由1.2.3.4映射为1.2.3.5

#### 1.2.10.6主体备用名称

主体备用名称的证书模板取值如下：

|  |  |  |
| --- | --- | --- |
| 名称 | 类型 | 说明 |
| name | 字符串 | 值必须为Subject Alternative Name |
| option | 字符串 | 如果不存在或者为fixed则所有的值都在证书模板。如果为optional则表示可选，其值在数据模板中，为required则表示必定存在该扩展，其值在数据模板中 |
| items | 对象的数组 | 数组中的每一项是个GeneralName |

GeneralName的证书模板取值如下：

|  |  |  |
| --- | --- | --- |
| 名称 | 类型 | 说明 |
| type | 字符串 | GeneralName的类型 |
| option | 字符串 | 如果不存在或者为fixed则所有的值都在证书模板。如果为optional则表示可选，其值在数据模板中，为required则表示必需，其值在数据模板中 |
| multiValue | 布尔类型 | 是否多值。不存在为单值 |
| value |  | 除directoryName外的其他类型的GeneralName值，option为fixed有效 |

数据模板在扩展2.5.29.17对应的项中，GeneralName的类型名对应的值通常就是GeneralName对应类型的值。

例如：

"extensions":{

"2.5.29.17":{

"rfc822Name":"test@test.com"

}

}

表示主体备用名的rfc822Name项的值为test@test.com

otherName的证书模板取值如下：

|  |  |  |
| --- | --- | --- |
| 名称 | 类型 | 说明 |
| type | 字符串 | 值必须为otherName |
| option | 字符串 | 如果不存在或者为fixed则所有的值都在证书模板。如果为optional则表示可选，其值在数据模板中，为required则表示必需，其值在数据模板中 |
| multiValue | 布尔类型 | 是否多值。不存在为单值 |
| otherNameType | 字符串 | 具体某一个otherName的OID |
| otherNameEncodeType | 字符串 | 值的编码类型。UTF8String、OctetString和whole |
| value | 字符串或者字符串数组 | option为fixed有效。字符串数组仅适用于多值。如果otherNameEncodeType为UTF8String则为实际的字符串值；为OctetString则为Hex编码的值；为whole则是值的DER编码的Hex编码 |

otherName的证书模板可以有多项，每一类otherName一项

otherName的数据模板是个对象数组，数组中的项取值如下：

|  |  |  |
| --- | --- | --- |
| 名称 | 类型 | 说明 |
| type | 字符串 | 具体某一个otherName的OID |
| value | 字符串或者字符串数组 | 同otherName的证书模板，但option为optional或者required其有效 |

如果type匹配otherName的证书模板的otherNameType，则使用该项的value。

例如：

证书模板

{

"name":"Subject Alternative Name",

"option":"required",

"items":[

{

"type":"otherName",

"otherNameType":"1.3.6.1.4.1.311.20.2.3",

"otherNameEncodeType":"UTF8String",

"option":"required"

}

]

}

对应的数据模板

"2.5.29.17":{

"otherName":[

{

"type":"1.3.6.1.4.1.311.20.2.3.1",

"value":"upn@test.com"

}

]

}

表示主体备用名有UPN,其值为upn@test.com

rfc822Name的证书模板取值如下：

|  |  |  |
| --- | --- | --- |
| 名称 | 类型 | 说明 |
| type | 字符串 | 值必须为rfc822Name |
| option | 字符串 | 如果不存在或者为fixed则所有的值都在证书模板。如果为optional则表示可选，其值在数据模板中，为required则表示必需，其值在数据模板中 |
| multiValue | 布尔类型 | 是否多值。不存在为单值 |
| value | 字符串或者字符串数组 | option为fixed有效。字符串数组仅适用于多值。Email值 |

例如：

证书模板

{

"name":"Subject Alternative Name",

"option":"required",

"items":[

{

"type":"rfc822Name",

" multiValue":true,

"option":"required"

}

]

}

对应的数据模板

"2.5.29.17":{

"rfc822Name":[

{

"test@test.com",

"test@你好.中国"

}

]

}

表示主体备用名有两个rfc822Name,其值分别为test@test.com和test@你好.中国

dNSName的证书模板取值如下：

|  |  |  |
| --- | --- | --- |
| 名称 | 类型 | 说明 |
| type | 字符串 | 值必须为dNSName |
| option | 字符串 | 如果不存在或者为fixed则所有的值都在证书模板。如果为optional则表示可选，其值在数据模板中，为required则表示必需，其值在数据模板中 |
| multiValue | 布尔类型 | 是否多值。不存在为单值 |
| value | 字符串或者字符串数组 | option为fixed有效。字符串数组仅适用于多值。DNS值 |

例如：

证书模板

{

"name":"Subject Alternative Name",

"option":"required",

"items":[

{

"type":"dNSName",

"option":"required"

}

]

}

对应的数据模板

"2.5.29.17":{

"dNSName":"test.com"

}

表示主体备用名有一个域名,其值为test.com

directoryName的证书模板取值如下：

|  |  |  |
| --- | --- | --- |
| 名称 | 类型 | 说明 |
| type | 字符串 | 值必须为directoryName |
| option | 字符串 | 如果不存在或者为fixed则所有的值都在证书模板。如果为optional则表示可选，其值在数据模板中，为required则表示必需，其值在数据模板中 |
| multiValue | 布尔类型 | 是否多值。不存在为单值 |
| template | 对象 | 目录名。同主体名 |

directoryName的数据模板的取值和主体名数据模板的取值一样

ediPartyName的证书模板取值如下：

|  |  |  |
| --- | --- | --- |
| 名称 | 类型 | 说明 |
| type | 字符串 | 值必须为ediPartyName |
| option | 字符串 | 如果不存在或者为fixed则所有的值都在证书模板。如果为optional则表示可选，其值在数据模板中，为required则表示必需，其值在数据模板中 |
| multiValue | 布尔类型 | 是否多值。不存在为单值 |
| value | 对象或者对象数组 | option为fixed有效。对象数组仅适用于多值。ediPartyName值 |

单个ediPartyName值的取值如下：

|  |  |  |
| --- | --- | --- |
| 名称 | 类型 | 说明 |
| nameAssigner | 对象 | 可选。DirectoryString类型 |
| partyName | 对象 | 必需。DirectoryString类型 |

DirectoryString类型的取值如下：

|  |  |  |
| --- | --- | --- |
| 名称 | 类型 | 说明 |
| encodeType | 字符串 | 编码类型，UTF8String等 |
| value | 字符串 | 字符串的值 |

uniformResourceIdentifier的证书模板取值如下：

|  |  |  |
| --- | --- | --- |
| 名称 | 类型 | 说明 |
| type | 字符串 | 值必须为uniformResourceIdentifier |
| option | 字符串 | 如果不存在或者为fixed则所有的值都在证书模板。如果为optional则表示可选，其值在数据模板中，为required则表示必需，其值在数据模板中 |
| multiValue | 布尔类型 | 是否多值。不存在为单值 |
| value | 字符串或者字符串数组 | option为fixed有效。字符串数组仅适用于多值。URI |

例如：

证书模板

{

"name":"Subject Alternative Name",

"option":"required",

"items":[

{

"type":"uniformResourceIdentifier",

"option":"required"

}

]

}

对应的数据模板

"2.5.29.17":{

"uniformResourceIdentifier":"http://test.com"

}

表示主体备用名有一个URI,其值为http://test.com

iPAddress的证书模板取值如下：

|  |  |  |
| --- | --- | --- |
| 名称 | 类型 | 说明 |
| type | 字符串 | 值必须为iPAddress |
| option | 字符串 | 如果不存在或者为fixed则所有的值都在证书模板。如果为optional则表示可选，其值在数据模板中，为required则表示必需，其值在数据模板中 |
| multiValue | 布尔类型 | 是否多值。不存在为单值 |
| value | 对象或者对象数组 | option为fixed有效。对象数组仅适用于多值。IP地址的值 |

单个IP地址的值如下表所示：

|  |  |  |
| --- | --- | --- |
| 名称 | 类型 | 说明 |
| encodeType | 字符串 | 值必须为hex |
| value | 字符串 | Hex编码的IP地址 |

例如：

证书模板

{

"name":"Subject Alternative Name",

"option":"required",

"items":[

{

"type":"iPAddress",

"option":"required"

}

]

}

对应的数据模板

"2.5.29.17":{

"iPAddress":{

"encodeType":"hex",

"value":"C0A80102"

}

}

表示主体备用名有一个IP地址,其值为192.168.1.2

registeredID的证书模板取值如下：

|  |  |  |
| --- | --- | --- |
| 名称 | 类型 | 说明 |
| type | 字符串 | 值必须为registeredID |
| option | 字符串 | 如果不存在或者为fixed则所有的值都在证书模板。如果为optional则表示可选，其值在数据模板中，为required则表示必需，其值在数据模板中 |
| multiValue | 布尔类型 | 是否多值。不存在为单值 |
| value | 字符串或者字符串数组 | option为fixed有效。字符串数组仅适用于多值。OID |

例如：

证书模板

{

"name":"Subject Alternative Name",

"option":"required",

"items":[

{

"type":"registeredID",

"option":"required"

}

]

}

对应的数据模板

"2.5.29.17":{

"registeredID":"1.2.156.1000"

}

表示主体备用名有一个注册ID,其值为1.2.156.1000

#### 1.2.10.7 颁发者备用名称

颁发者备用名称和主体备用名称的模板基本一样。只是其name为Issuer Alternative Name，option必须为fixed，不存在数据模板。

颁发者备用名称的证书模板取值如下：

|  |  |  |
| --- | --- | --- |
| 名称 | 类型 | 说明 |
| name | 字符串 | 值必须为Issuer Alternative Name |
| option | 字符串 | 不存在或者必须为fixed |
| items | 对象的数组 | 数组中的每一项是个GeneralName。但里面的每个名称的option也只能为fixed或者省略 |

#### 1.2.10.8主体目录属性

主体目录属性的证书模板取值如下：

|  |  |  |
| --- | --- | --- |
| 名称 | 类型 | 说明 |
| name | 字符串 | 值必须为Subject Alternative Name |
| option | 字符串 | 如果不存在或者为fixed则所有的值都在证书模板。如果为optional则表示可选，其值在数据模板中，为required则表示必定存在该扩展，其值在数据模板中 |
| items | 对象的数组 | 数组中的每一项是个属性，和主体的属性一样 |

主体目录属性的数据模板也和主体的数据模板一样

例如：

证书模板

{

"name":"Subject Directory Attributes",

"option":"required",

"items":[

{

"option":"fixed",

"attrType":"C",

"encodeType":"PrintableString",

"attrValue":"CN"

},

{

"option":"optional",

"attrType":"O",

"encodeType":"UTF8String",

"multiValue":true

},

{

"option":"required",

"attrType":"CN",

"encodeType":"UTF8String"

}

]

}

对应的数据模板

"2.5.29.9":{

"2.5.4.3":"通用名称",

"2.5.4.10":["组织名","O"]

}

表示主体目录属性有三个属性，一个是C,其值为CN。一个是O，这是个多值属性，其值为组织名和O。一个是CN,其值为通用名称

#### 1.2.10.9 基本约束

基本约束只有证书模板，没有数据模板

基本约束的证书模板取值如下：

|  |  |  |
| --- | --- | --- |
| 名称 | 类型 | 说明 |
| name | 字符串 | 值必须为Basic Constraints |
| value | 对象 | 基本约束的值 |

基本约束的值取值如下：

|  |  |  |
| --- | --- | --- |
| 名称 | 类型 | 说明 |
| cA | 布尔类型 | 是否为CA |
| pathLenConstraint | 数字（整数） | 可选。在cA为true时有效 |

例如：

{

"name":"Basic Constraints",

"critical":true,

"value":{

"cA":true

}

}

表示一个关键的基本约束扩展，其cA项为真，下面可以签证书层级不限

#### 1.2.10.10 名称约束

名称约束只有证书模板，没有数据模板

名称约束的证书模板取值如下：

|  |  |  |
| --- | --- | --- |
| 名称 | 类型 | 说明 |
| name | 字符串 | 值必须为Name Constraints |
| value | 对象 | 名称约束的值 |

名称约束的值取值如下：

|  |  |  |
| --- | --- | --- |
| 名称 | 类型 | 说明 |
| permittedSubtrees | 对象的数组 | 可选。每个对象为GeneralSubtree类型 |
| excludedSubtrees | 对象的数组 | 可选。每个对象为GeneralSubtree类型 |

GeneralSubtree类型取值如下：

|  |  |  |
| --- | --- | --- |
| 名称 | 类型 | 说明 |
| base | 对象 | GeneralName类型，见主体备用名称 |
| minimum | 整数 | 可选 |
| maximum | 整数 | 可选 |

例如：

{

"name":"Name Constraints",

"critical":true,

"value":{

"excludedSubtrees":[

{

"base":{

"type":"rfc822Name",

"option":"fixed",

"value":"abc.test"

}

}

]

}

}

表示一个关键的名称约束扩展，其有一个禁止子树，禁止rfc822Name的abc.test

#### 1.2.10.11策略约束

策略约束只有证书模板，没有数据模板

策略约束的证书模板取值如下：

|  |  |  |
| --- | --- | --- |
| 名称 | 类型 | 说明 |
| name | 字符串 | 值必须为Policy Constraints |
| value | 对象 | 策略约束的值 |

策略约束的值取值如下：

|  |  |  |
| --- | --- | --- |
| 名称 | 类型 | 说明 |
| requireExplicitPolicy | 整数 | 可选。 |
| inhibitPolicyMapping | 整数 | 可选。 |

例如：

{

"name":"Policy Constraints",

"critical":true,

"value":{

"requireExplicitPolicy":1

}

}

表示一个关键的策略约束，其跳过1个证书后要求显式的证书策略

#### 1.2.10.12 扩展密钥用法

扩展密钥用法只有证书模板，没有数据模板

扩展密钥用法的证书模板取值如下：

|  |  |  |
| --- | --- | --- |
| 名称 | 类型 | 说明 |
| name | 字符串 | 值必须为Extended Key Usage |
| value | 字符串数组 | 每个字符串表示一个扩展密钥用法的OID |

例如：

{

"name":"Extended Key Usage",

"value":[

"1.2.3.4.5.6.7.8",

"1.3.6.1.5.5.7.3.1"

]

}

表示一个扩展密钥用法扩展，其有两个扩展密钥用法，分别为1.2.3.4.5.6.7.8和1.3.6.1.5.5.7.3.1

#### 1.2.10.13 CRL发布点

CRL发布点只有证书模板，没有数据模板

CRL发布点的证书模板取值如下：

|  |  |  |
| --- | --- | --- |
| 名称 | 类型 | 说明 |
| name | 字符串 | 值必须为CRL Distribution Points |
| value | 对象数组 | 每个对象是DistributionPoint类型 |

DistributionPoint类型的取值如下：

|  |  |  |
| --- | --- | --- |
| 名称 | 类型 | 说明 |
| distributionPoint | 对象 | 可选。DistributionPointName类型 |
| reasons | 对象 | 可选。 |
| cRLIssuer | 对象数组 | 每个元素是GeneralName类型，见主体备用名称 |

DistributionPointName类型的取值如下：

|  |  |  |
| --- | --- | --- |
| 名称 | 类型 | 说明 |
| type | 字符串 | 值为fullName或者nameRelativeToCRLIssuer |
| value |  | 根据type而定。 |

fullName的值是个对象数组。每个元素是GeneralName类型

nameRelativeToCRLIssuer的值是个对象，取值如下：

|  |  |  |
| --- | --- | --- |
| 名称 | 类型 | 说明 |
| attrType | 字符串 | 属性类型的名称或者OID |
| encodeType | 字符串 | 编码类型，例如：UTF8String |
| attrValue | 字符串 | 属性值 |

reasons对象取值如下：

|  |  |  |
| --- | --- | --- |
| 名称 | 类型 | 说明 |
| encodeType | 字符串 | 值为name或者bit |
| value | 数组 | 如果是name，则每个元素为字符串类型，表示作废原因的名称（比如：keyCompromise）。如果是bit则每个元素是布尔类型，表示按位置设置置位的值 |

例如：

{

"name":"CRL Distribution Points",

"critical":false,

"value":[

{

"distributionPoint":{

"type":"fullName",

"value":[

{

"type":"uniformResourceIdentifier",

"option":"fixed",

"value":"http://www.tesr.com/crl"

}

]

}

}

]

}

表示一个CRL发布点扩展，其有一个发布点，发布点名使用fullName类型，其URI为<http://www.tesr.com/crl>

#### 1.2.10.14 禁止任意策略

禁止任意策略只有证书模板，没有数据模板

禁止任意策略的证书模板取值如下：

|  |  |  |
| --- | --- | --- |
| 名称 | 类型 | 说明 |
| name | 字符串 | 值必须为Inhibit anyPolicy |
| value | 整数 | 禁止任意策略的值 |

例如：

{

"name":"Inhibit anyPolicy",

"critical":true,

"encodeType":"default",

"value":1

}

表示一个关键的禁止任意策略的扩展，其跳过一个证书后禁止任意策略OID。

#### 1.2.10.15 Freshest CRL

Freshest CRL和CRL发布点一样，除了name为Freshest CRL之外。

#### 1.2.10.16 Authority Information Access

Authority Information Access只有证书模板，没有数据模板

Authority Information Access的证书模板取值如下：

|  |  |  |
| --- | --- | --- |
| 名称 | 类型 | 说明 |
| name | 字符串 | 值必须为Authority Information Access |
| value | 对象数组 | 每个对象是AccessDescription类型 |

AccessDescription类型取值如下：

|  |  |  |
| --- | --- | --- |
| 名称 | 类型 | 说明 |
| accessMethod | 字符串 | 访问方法的名称或者OID |
| accessLocation | 对象 | GeneralName类型 |

例如：

{

"name":"Authority Information Access",

"value":[

{

"accessMethod":"ocsp",

"accessLocation":{

"type":"uniformResourceIdentifier",

"option":"fixed",

"value":"http://test.com/ocsp"

}

},

{

"accessMethod":"caIssuers",

"accessLocation":{

"type":"uniformResourceIdentifier",

"option":"fixed",

"value":"http://test.com/ca.p7c"

}

}

]

}

表示有AIA扩展。AIA扩展包括OCSP地址为<http://test.com/ocsp，CA>证书的地址为<http://test.com/ca.p7c>

#### 1.2.10.17 Subject Information Access

Subject Information Access和Authority Information Access一样，除了name为Subject Information Access之外

#### 1.2.10.17 OCSP NoCheck

OCSP NoCheck只有证书模板，没有数据模板

OCSP NoCheck的证书模板取值如下：

|  |  |  |
| --- | --- | --- |
| 名称 | 类型 | 说明 |
| name | 字符串 | 值必须为OCSP NoCheck |

例如：

{

"name":"OCSP NoCheck"

}

表示一个非关键的OCSP NoCheck扩展

#### 1.2.10.18 私钥使用期

私钥使用期的证书模板取值如下：

|  |  |  |
| --- | --- | --- |
| 名称 | 类型 | 说明 |
| name | 字符串 | 值必须为Private key usage period |
| option | 字符串 | 如果不存在或者为fixed则所有的值都在证书模板。如果为optional则表示可选，其值在数据模板中，为required则表示必定存在该扩展，其值在数据模板中 |
| template | 对象 |  |

template的取值如下：

|  |  |  |
| --- | --- | --- |
| 名称 | 类型 | 说明 |
| notBefore | 对象 | 可选。时间类型 |
| notAfter | 对象 | 可选。时间类型 |

时间类型的取值如下：

|  |  |  |
| --- | --- | --- |
| 名称 | 类型 | 说明 |
| external | 布尔类型 | 为真表示值在数据模板 |
| value | 字符串 | 时间的值。External为false时有效，格式为YYYYMMDDhhmmssZ |

对应的数据模板的取值如下：

|  |  |  |
| --- | --- | --- |
| 名称 | 类型 | 说明 |
| notBefore | 字符串 | 时间的值。格式为YYYYMMDDhhmmssZ |
| notAfter | 字符串 | 时间的值。格式为YYYYMMDDhhmmssZ |

例如：

证书模板

{

"name":"Private key usage period",

"option":"optional",

"template":{

"notBefore":{

"external":true

},

"notAfter":{

"external":true

}

}

}

对应的数据模板为

"2.5.29.16":{

"notBefore":"20190102020309Z",

"notAfter":"20990102020309Z"

}

表示一个私钥使用期扩展，其notBefore为UTC时间2019年1月2日2点3分9秒，notAfter为UTC时间2099年1月2日2点3分9秒

#### 1.2.10.19 其他扩展

其他扩展的证书模板取值如下：

|  |  |  |
| --- | --- | --- |
| 名称 | 类型 | 说明 |
| name | 字符串 | 扩展的OID |
| option | 字符串 | 如果不存在或者为fixed则所有的值都在证书模板。如果为optional则表示可选，其值在数据模板中，为required则表示必定存在该扩展，其值在数据模板中 |
| encodeType | 字符串 | 编码类型。可以为whole、UTF8String、OctetString |
| value | 字符串 | 可选。Option为fixed有效对于whole为整个ASN.1结构的DER编码再Hex编码，对于UTF8String就是字符串本身，对于OctetString则是Hex编码的值 |

例如：

证书模板

{

"name":"1.2.3.400",

"encodeType":"whole",

"value":"020101"

}

表示非关键扩展1.2.3.400，其类型为整数，值为1

证书模板

{

"name":"1.2.3.401",

"encodeType":"UTF8String",

"option":"required"

}

对应的数据模板

"1.2.3.401":"自定义的值"

表示非关键扩展1.2.3.401，其类型为UTF8String，值为自定义的值

# 2 签CRL

## 2.1 NetcaPKICertSignCert

原型：

int NETCADLLAPI NetcaPKICertSignCRL(NETCA\_PKI\_CERTIFICATE\_HANDLE issuerCert,const char \*templateInfo,void \*context,NetcaSignCallback\_Ptr signCallback,unsigned char\*\* pCrl,int \*pCrlLen)

参数：

issuerCert：颁发者证书，不能为NULL

templateInfo：输入参数，模板信息，UTF-8编码的JSON格式的对象

context：signCallback函数使用的上下文信息

signCallback：回调的签名函数，如果不为NULL，则使用该回调函数进行签名，否则使用issuerCert关联的私钥进行签名。

pCrl：输出参数，成功返回签发出的CRL的编码

pCrlLen：输出参数，成功返回签发出的CRL的编码的字节数

返回值：

成功返回NETCA\_PKI\_SUCCESS，如果是templateInfo有问题返回NETCA\_PKI\_TEMPLATE\_FAIL。出错返回其他值。

说明：

根据模板来签CRL。模板使用JSON来表示，详见2.2。

本函数没有对模板进行严格检查，可能会签发出一些不符合标准的CRL，因此保证模板的正确性就比较关键。

## 2.2 模板说明

### 2.2.1版本号

版本号总是使用版本2，模板中不能进行相关设置。

### 2.2.2签名算法

签名算法的模板为signatureAlgorithm，对应的类型为对象。其和1.2.3证书的签名算法一样。

### 2.2.3颁发者名

颁发者名总是取颁发者证书的主体名，模板中不能进行相关设置。

### 2.2.4生效日期

生效日期的模板为thisUpdate，对应的类型为对象。有两种方式来设置：

1. 使用当前时间

不使用thisUpdate项。或者证书模板的取值如下：

|  |  |  |
| --- | --- | --- |
| 名称 | 类型 | 说明 |
| type | 字符串 | 值必须为now |

1. 使用固定的生效时间

证书模板的取值如下：

|  |  |  |
| --- | --- | --- |
| 名称 | 类型 | 说明 |
| type | 字符串 | 值必须为time |
| value | 字符串 | 时间，格式为YYYYMMDDhhmmssZ |

例如：

"thisUpdate":{

"type":"time",

"value":"20180501020314Z"

}

表示设置有效期开始时间为UTC时间:2018年5月1日2点3分14秒。

### 2.2.5下一次更新的时间

下一次更新的时间的模板为nextUpdate，对应的类型为对象。有三种方式来设置：

1. 不使用下一次更新的时间

通常不应该不使用下一次更新的时间。如果真要这样用，直接省略nextUpdate即可

2、设置下一次更新的时间为有生效日期加上某个值。比如说，有效期为一年，也就是下一次更新的时间为生效日期加上1年。

证书模板的取值如下：

|  |  |  |
| --- | --- | --- |
| 名称 | 类型 | 说明 |
| type | 字符串 | 值必须为interval |
| value | 对象 | 有效期的时间 |

value的取值，如下表所示：

|  |  |  |
| --- | --- | --- |
| 名称 | 类型 | 说明 |
| year | 数字(整数) | 年，可选 |
| month | 数字(整数) | 月，可选 |
| day | 数字(整数) | 日，可选 |
| hour | 数字(整数) | 小时，可选 |

例如：

"nextUpdate":{

"type":"interval",

"value":{

"day":7

}

}

表示下一次更新的时间为生效日期后7天

3、设置下一次更新的时间为某个固定的时间

证书模板的取值如下：

|  |  |  |
| --- | --- | --- |
| 名称 | 类型 | 说明 |
| type | 字符串 | 值必须为time |
| value | 字符串 | 时间，格式为YYYYMMDDhhmmssZ |

例如：

"nextUpdate":{

"type":"time",

"value":"20180904051627Z"

}

表示下一次更新的时间为UTC时间2018年9月4日5点16分27秒

### 2.2.6作废证书列表

作废证书列表的时间的模板为revokedCertificates，对应的类型为数组。数组中每个元素为对象类型，表示一个吊销证书。

如果不存在作废证书，则省略revokedCertificates即可。

每个作废证书的取值如下：

|  |  |  |
| --- | --- | --- |
| 名称 | 类型 | 说明 |
| serialNumber | 字符串 | 作废证书的序列号的值的Hex编码 |
| revocationDate | 字符串 | 作废时间，格式为YYYYMMDDhhmmssZ |
| extensions | 对象数组 | CRL条目扩展，可选 |

例如：

{

"serialNumber":"1234",

"revocationDate":"20180102135427Z"

}

表示一个作废证书，其序列号为"\x12\x34",作废时间为UTC时间2018年1月2日13点54分27秒，没有相关的扩展。

每个CRL条目扩展为一个对象，扩展有以下公用的项

|  |  |  |
| --- | --- | --- |
| 名称 | 类型 | 说明 |
| name | 字符串 | 扩展的名称或者扩展的OID |
| critical | 布尔类型 | 为真表示关键扩展，为假表示非关键扩展。不存在表示非关键扩展 |

下面逐个说明各个扩展的模板

#### 2.2.6.1 作废原因

作废原因有两种方式：

1、使用名称

模板的取值如下：

|  |  |  |
| --- | --- | --- |
| 名称 | 类型 | 说明 |
| name | 字符串 | 值必须为Reason Code |
| value | 字符串 | 作废原因的名称 |

例如：

{

"name":"Reason Code",

"value":"keyCompromise"

}

表示作废原因扩展，作废原因为密钥泄露

2、直接写枚举的值

证书模板的取值如下：

|  |  |  |
| --- | --- | --- |
| 名称 | 类型 | 说明 |
| name | 字符串 | 值必须为Reason Code |
| value | 整数 | 作废原因的枚举值 |

例如：

{

"name":"Reason Code",

"value":6

}

表示作废原因扩展，作废原因为证书挂起

#### 2.2.6.2 无效时间

无效时间模板的取值如下：

|  |  |  |
| --- | --- | --- |
| 名称 | 类型 | 说明 |
| name | 字符串 | 值必须为Invalidity Date |
| value | 字符串 | 无效时间，格式为YYYYMMDDhhmmssZ |

例如：

{

"name":"Invalidity Date",

"value":"20180101135427Z"

}

表示无效时间扩展，无效时间为UTC时间2018年1月1日13点54分27秒

#### 2.2.6.3 Hold Instruction Code

Hold Instruction Code模板的取值如下：

|  |  |  |
| --- | --- | --- |
| 名称 | 类型 | 说明 |
| name | 字符串 | 值必须为Hold Instruction Code |
| value | 字符串 | 必须为OID |

例如：

{

"name":"Hold Instruction Code",

"value":"1.2.840.10040.2.3"

}

表示Hold Instruction Code扩展，指令为拒绝

#### 2.2.6.4证书颁发者

证书颁发者模板的取值如下：

|  |  |  |
| --- | --- | --- |
| 名称 | 类型 | 说明 |
| name | 字符串 | 值必须为Hold Instruction Code |
| value | 对象 | directoryName名称的值为颁发者名的DER编码的HEX字符串 |

例如：

{

"name":"Certificate Issuer",

"critical":true,

"value":{

"directoryName":"302A310B300906035504061302434E311B301906035504030C12E68891E698AFE6A0B9E68891E68095E8B081"

}

}

表示关键的证书颁发者扩展，其使用了一个directoryName类型的名称，其编码为302A310B300906035504061302434E311B301906035504030C12E68891E698AFE6A0B9E68891E68095E8B081

#### 2.2.6.5其他扩展

同1.2.10.19证书的其他扩展，只是其值必须在模板中填写

### 2.2.7 CRL扩展

CRL扩展是extensions项，这是一个对象数组，数组中每一项表示一个CRL扩展。

每个扩展有以下公用的项

|  |  |  |
| --- | --- | --- |
| 名称 | 类型 | 说明 |
| name | 字符串 | 扩展的名称或者扩展的OID |
| critical | 布尔类型 | 为真表示关键扩展，为假表示非关键扩展。不存在表示非关键扩展 |

下面逐个说明各个扩展的模板

#### 2.2.7.1 颁发者密钥标识符

同1.2.10.1证书的颁发者密钥标识符扩展

#### 2.2.7.2 颁发者备用名称

同1.2.10.7证书的颁发者备用名称扩展

#### 2.2.7.3 CRL数字

CRL数字的模板如下表所示：

|  |  |  |
| --- | --- | --- |
| 名称 | 类型 | 说明 |
| name | 字符串 | 值必须为CRL Number |
| value | 字符串 | Hex编码的整数编码 |

例如：

{

"name":"CRL Number",

"value":"00ac1234"

}

表示CRL数字扩展，其值为"\x00\xac\x12\x34"。如果值为"ac1234"也是同样编码，总是把它当作非负数处理的

#### 2.2.7.4 增量CRL指示器

增量CRL指示器的模板如下表所示：

|  |  |  |
| --- | --- | --- |
| 名称 | 类型 | 说明 |
| name | 字符串 | 值必须为Delta CRL Indicator |
| value | 字符串 | Hex编码的整数编码 |

例如：

{

"name":"Delta CRL Indicator",

"critical":true,

"value":"00ac1233"

}

表示关键的增量CRL指示器扩展，其基准CRL的CRL数字为"\x00\xac\x12\x33"。如果值为" ac1233"也是同样编码，总是把它当作非负数处理的

#### 2.2.7.5 Issuing Distribution Point

Issuing Distribution Point的模板如下表所示：

|  |  |  |
| --- | --- | --- |
| 名称 | 类型 | 说明 |
| name | 字符串 | 值必须为Issuing Distribution Point |
| value | 对象 |  |

value的取值如下表所示：

|  |  |  |
| --- | --- | --- |
| 名称 | 类型 | 说明 |
| distributionPoint | 对象 | 可选，同CRL发布点的对应对象 |
| onlyContainsUserCerts | 布尔值 | 可选，不存在为false |
| onlyContainsCACerts | 布尔值 | 可选，不存在为false |
| onlySomeReasons | 对象 | 可选，同CRL发布点的对应对象 |
| indirectCRL | 布尔值 | 可选，不存在为false |
| onlyContainsAttributeCerts | 布尔值 | 可选，不存在为false |

例如：

{

"name":"Issuing Distribution Point",

"critical":true,

"value":{

"onlyContainsUserCerts":true

}

}

表示关键的Issuing Distribution Point扩展，该CRL仅仅包含用户证书

例如：

{

"name":"Issuing Distribution Point",

"critical":true,

"value":{

"onlySomeReasons":{

"encodeType":"name",

"value":[

"keyCompromise","cACompromise"

]

}

}

}

表示关键的Issuing Distribution Point扩展，该CRL仅覆盖密钥泄露和CA密钥泄露两种作废原因

例如：

{

"name":"Issuing Distribution Point",

"critical":true,

"value":{

"distributionPoint":{

"type":"fullName",

"value":[

{

"type":"uniformResourceIdentifier",

"option":"fixed",

"value":"http://www.test.com/crl"

}

]

}

}

}

表示关键的Issuing Distribution Point扩展，该CRL的发布点为URI类型的名称，值为<http://www.test.com/crl>

#### 2.2.7.6 Freshest CRL

同1.2.10.15证书的Freshest CRL扩展

#### 2.2.7.7 Authority Information Access

同1.2.10.16证书的Authority Information Access扩展

#### 2.2.7.8 AA issuing distribution point

AA issuing distribution point的模板如下表所示：

|  |  |  |
| --- | --- | --- |
| 名称 | 类型 | 说明 |
| name | 字符串 | 值必须为AA issuing distribution point |
| value | 对象 |  |

value的取值如下表所示：

|  |  |  |
| --- | --- | --- |
| 名称 | 类型 | 说明 |
| distributionPoint | 对象 | 可选，同CRL发布点的对应对象 |
| onlySomeReasons | 对象 | 可选，同CRL发布点的对应对象 |
| indirectCRL | 布尔值 | 可选，不存在为false |
| containsUserAttributeCerts | 布尔值 | 可选，不存在为true |
| containsAACerts | 布尔值 | 可选，不存在为true |
| containsSOAPublicKeyCerts | 布尔值 | 可选，不存在为true |

例如；

{

"name":"AA issuing distribution point",

"critical":true,

"value":{

"onlySomeReasons":{

"encodeType":"bit",

"value":[

false,true,true

]

}

}

}

表示关键的AA issuing distribution point扩展，该CRL仅覆盖密钥泄露和CA密钥泄露两种作废原因

#### 2.2.7.9其他扩展

同1.2.10.19证书的其他扩展，只是其值必须在模板中填写

# 3 OCSP

## 3.1 NetcaPKICertOCSPRequest2Json

原型：

int NETCADLLAPI NetcaPKICertOCSPRequest2Json(const unsigned char \*reqEncode,int reqEncodeLen,char \*\*pJsonEncode,int \*pJsonEncodeLen)

参数：

reqEncode：输入参数，OCSPRequest的DER编码

reqEncodeLen：输入参数，OCSPRequest的DER编码字节数

pJsonEncode：输出参数，成功返回UTF-8编码的JSON数据

pJsonEncodeLen：输出参数，成功返回\*pJsonEncode的字节数

返回值：

成功返回NETCA\_PKI\_SUCCESS出错返回其他值。

说明：

把OCSPRequest的DER编码转换为JSON对象的编码，方便使用JSON来进行处理

## 3.2 OCSPRequest的JSON对象

OCSPRequest的JSON对象，取值如下：

|  |  |  |
| --- | --- | --- |
| 名称 | 类型 | 说明 |
| tbsRequest | 对象 | 必需，OCSP请求信息 |
| tbsRequestEncode | 字符串 | 可选，tbsRequest的DER编码后再Hex编码。如果optionalSignature存在则存在 |
| optionalSignature | 对象 | 可选，签名 |

### 3.2.1 OCSPRequest的tbsRequest

OCSPRequest的tbsRequest的取值如下：

|  |  |  |
| --- | --- | --- |
| 名称 | 类型 | 说明 |
| version | 整数或者字符串 | 必需，如果是小整数，则为整数类型，如果是大整数，则为Hex编码的字符串 |
| requestorName | 对象 | 可选，请求者名 |
| requestList | 对象数组 | 必需，每个元素为一个证书的请求 |
| extensions | 对象数组 | 可选，每个元素为一个扩展 |

正常情况下，version应该为0。

如果没签名，则requestorName通常也不会存在

requestorName如果是目录名类型，则取值如下：

|  |  |  |
| --- | --- | --- |
| 名称 | 类型 | 说明 |
| value | 对象 | 必需，directoryName名称的对应的值为字符串类型，表示Hex编码的DN编码 |

如果是其他类型，则取值如下：

|  |  |  |
| --- | --- | --- |
| 名称 | 类型 | 说明 |
| value | 字符串 | 必需，整个GeneralName的DER编码的Hex编码 |
| encodeType | 字符串 | 必须为whole |

requestList中的每个请求的取值如下：

|  |  |  |
| --- | --- | --- |
| 名称 | 类型 | 说明 |
| reqCert | 对象 | 必需，CertID类型 |
| extensions | 对象数组 | 可选，每个元素为一个扩展 |

CertID类型的取值如下：

|  |  |  |
| --- | --- | --- |
| 名称 | 类型 | 说明 |
| hashAlgorithm | 对象 | 必需，Hash算法 |
| issuerNameHash | 字符串 | 必需，Hex编码 |
| issuerKeyHash | 字符串 | 必需，Hex编码 |
| serialNumber | 字符串 | 必需，Hex编码 |

Hash算法的取值如下：

|  |  |  |
| --- | --- | --- |
| 名称 | 类型 | 说明 |
| algorithm | 字符串 | 必需，OID |
| parameters | 字符串 | 可选，DER编码的Hex编码 |

OCSPRequest的tbsRequest的扩展和requestList中的每个请求的扩展的类型是一样的，取值如下：

|  |  |  |
| --- | --- | --- |
| 名称 | 类型 | 说明 |
| extnID | 字符串 | 必需，扩展的OID |
| critical | 布尔值 | 必需，是否关键扩展 |
| extnValue | 字符串 | 必需，扩展值的Hex编码 |

例如：

{

"tbsRequest":{

"version":0,

"requestList":

[

{

"reqCert":{

"hashAlgorithm":{"algorithm":"1.3.14.3.2.26"},

"issuerNameHash":"454A64FE7FC64AF40FAB42F2F660C39AEC932A30",

"issuerKeyHash":"270E455DB5473A5E9BEE8F814472AC051BE34B03",

"serialNumber":"108ED77483B8B42B754129"

}

}

],

"extensions":

[

{

"extnID":"1.3.6.1.5.5.7.48.1.2",

"critical":false,

"extnValue":"04147C5968535B2D63D1E2D15D426C97F0E67A74D72E"

}

]

}

}

### 3.2.2 OCSPRequest的optionalSignature

OCSPRequest的optionalSignature的取值如下：

|  |  |  |
| --- | --- | --- |
| 名称 | 类型 | 说明 |
| signatureAlgorithm | 对象 | 必需，签名算法，同CertID类型的Hash算法 |
| signature | 对象 | 必需，签名值 |
| certs | 字符串对象 | 可选，每个元素是Base64编码的证书，不包括头和尾也不包括回车换行 |

签名值的取值如下：

|  |  |  |
| --- | --- | --- |
| 名称 | 类型 | 说明 |
| unusedBits | 整数 | 必需，未使用位。0-7 |
| value | 字符串 | 必需，Hex编码的值 |

## 3.3 NetcaPKICertSignOCSPResponse

原型：

int NETCADLLAPI NetcaPKICertSignOCSPResponse(NETCA\_PKI\_CERTIFICATE\_HANDLE ocspCert,const char \*templateInfo,void \*context,NetcaSignCallback\_Ptr signCallback,unsigned char\*\* pResp,int \*pRespLen)

参数：

ocspCert：OCSP证书

templateInfo：输入参数，模板数据，UTF-8编码的JSON对象

context：signCallback函数使用的上下文信息

signCallback：回调的签名函数，如果不为NULL，则使用该回调函数进行签名，否则使用ocspCert关联的私钥进行签名。

pResp：输出参数，OCSPResponse的DER编码

pRespLen：输出参数，OCSPResponse的DER编码字节数

返回值：

成功返回NETCA\_PKI\_SUCCESS出错返回其他值。

说明：

根据模板签OCSPResponse

## 3.4 OCSPResponse模板

OCSPResponse模板是个对象，其取值如下：

|  |  |  |
| --- | --- | --- |
| 名称 | 类型 | 说明 |
| tbsResponseData | 对象 | 必需，要签名的响应数据 |
| signatureAlgorithm | 对象 | 必需，签名算法，同证书的签名算法 |
| certs | 字符串数组 | 可选，每个元素是Base64编码的证书，不包括头和尾也不包括回车换行 |

tbsResponseData的取值如下：

|  |  |  |
| --- | --- | --- |
| 名称 | 类型 | 说明 |
| version | 证书 | 可选，版本号，默认为0。 |
| responderID | 字符串 | 必需，name为使用名称，key为使用公钥的Hash |
| producedAt | 字符串 | 必需，产生时间，格式为YYYYMMDDhhmmssZ |
| responses | 对象数组 | 必须，每个元素为一个证书的状态的响应 |
| extensions | 对象数组 | 可选。每个元素是一个扩展，同证书的其他扩展 |

responses的每个元素的取值如下：

|  |  |  |
| --- | --- | --- |
| 名称 | 类型 | 说明 |
| certID | 对象 | 必需，同OCSPRequest的CertID。 |
| certStatus | 字符串 | 必需，取值为good、revoked、unknown三者之一 |
| revokedInfo | 对象 | 可选，作废时必需。 |
| thisUpdate | 字符串 | 必需，格式为YYYYMMDDhhmmssZ |
| nextUpdate | 对象 | 可选 |
| extensions | 对象数组 | 可选。每个元素是一个扩展。同CRL的作废证书列表扩展 |

revokedInfo的取值如下：

|  |  |  |
| --- | --- | --- |
| 名称 | 类型 | 说明 |
| revocationTime | 字符串 | 必需，作废时间，格式为YYYYMMDDhhmmssZ |
| revocationReason | 字符串或者整数 | 必需，同CRL的作废原因的value |

nextUpdate的取值如下：

|  |  |  |
| --- | --- | --- |
| 名称 | 类型 | 说明 |
| type | 字符串 | 必需，取值为time、interval两者之一。 |
| value | 字符串或者对象 | 必需，如果为time则是字符串格式为 YYYYMMDDhhmmssZ。否则为对象，同CRL的下一次更新的时间 |

# 4 时间戳

## 4.1 NetcaPKIMsgTimeStampReq2Json

原型：

int NETCADLLAPI NetcaPKIMsgTimeStampReq2Json(const unsigned char \*reqEncode,int reqEncodeLen,char \*\*pJsonEncode,int \*pJsonEncodeLen)

参数：

reqEncode：输入参数，TimeStampReq 的DER编码

reqEncodeLen：输入参数，TimeStampReq的DER编码字节数

pJsonEncode：输出参数，成功返回UTF-8编码的JSON数据

pJsonEncodeLen：输出参数，成功返回\*pJsonEncode的字节数

返回值：

成功返回NETCA\_PKI\_SUCCESS出错返回其他值。

说明：

把TimeStampReq的DER编码转换为JSON对象的编码，方便使用JSON来进行处理

## 4.2 TimeStampReq的JSON对象

TimeStampReq的取值如下：

|  |  |  |
| --- | --- | --- |
| 名称 | 类型 | 说明 |
| version | 整数或者字符串 | 必需，如果是小整数，则为整数类型，如果是大整数，则为Hex编码的字符串 |
| messageImprint | 对象 | 必需 |
| reqPolicy | 布尔值 | 可选，请求的策略OID |
| nonce | 字符串 | 可选，Hex编码的字符串 |
| certReq | 布尔值 | 必需，是否要求响应带证书 |
| extensions | 对象数组 | 可选，每个元素为一个扩展，同证书的其他扩展 |

messageImprint的取值如下：

|  |  |  |
| --- | --- | --- |
| 名称 | 类型 | 说明 |
| hashAlgorithm | 对象 | 必需，同OCSP的CertID的hashAlgorithm |
| hashedMessage | 字符串 | 必需，Hash值的Hex编码 |

## 4.3 NetcaPKIMsgJson2TSTInfo

原型：

int NETCADLLAPI NetcaPKIMsgJson2TSTInfo(const char \*json,unsigned char \*\*pTSTInfoEncode,int \*pTSTInfoEncodeLen)

参数：

json：输入参数，JSON的UTF-8编码

pTSTInfoEncode：输出参数，成功返回TSTInfo的DER编码

pTSTInfoEncodeLen：输出参数，成功返回\*pTSTInfoEncode的字节数

返回值：

成功返回NETCA\_PKI\_SUCCESS出错返回其他值。

说明：

根据JSON编码转换为TSTInfo的DER编码

## 4.4 TSTInfo的JSON对象

TSTInfo的JSON对象，取值如下：

|  |  |  |
| --- | --- | --- |
| 名称 | 类型 | 说明 |
| policy | 字符串 | 必需，时间戳策略OID |
| messageImprint | 对象 | 必需，同请求的messageImprint |
| serialNumber | 字符串 | 必需，时间戳序列号的Hex编码 |
| genTime | 字符串 | 可选，格式为YYYYMMDDhhmmssZ。默认为当前时间 |
| accuracy | 对象 | 可选。时间精度 |
| ordering | 布尔值 | 可选，默认为false |
| nonce | 字符串 | 可选，Hex编码 |
| tsaName | 对象 | 可选，时间戳的名称 |
| extensions | 对象数组 | 每个元素为一个扩展，同证书的其他扩展 |

accuracy的取值如下:

|  |  |  |
| --- | --- | --- |
| 名称 | 类型 | 说明 |
| seconds | 整数 | 可选，秒 |
| millis | 整数 | 可选，毫秒 |
| micros | 整数 | 可选，微秒 |

tsaName如果是目录名取值如下：

|  |  |  |
| --- | --- | --- |
| 名称 | 类型 | 说明 |
| value | 对象 | directoryName名称的值为字符串表示DN的DER编码的Hex编码 |

tsaName如果是其他类型，取值如下：

|  |  |  |
| --- | --- | --- |
| 名称 | 类型 | 说明 |
| encodeType | 字符串 | 必须为whole |
| value | 字符串 | 整个的GeneralName  的DER编码的Hex编码 |

## 4.5 NetcaPKIMsgJson2TimeStampResp

原型：

int NETCADLLAPI NetcaPKIMsgJson2TimeStampResp(const char \*json,unsigned char \*\*pTimeStampRespEncode,int \*pTimeStampRespEncodeLen)

参数：

json：输入参数，JSON的UTF-8编码

pTimeStampRespEncode：输出参数，成功返回TimeStampResp的DER编码

pTimeStampRespEncodeLen：输出参数，成功返回\*pTimeStampRespEncode的字节数

返回值：

成功返回NETCA\_PKI\_SUCCESS出错返回其他值。

说明：

根据JSON编码转换为TimeStampResp的DER编码

## 4.6 TimeStampResp的JSON对象

TimeStampResp的JSON对象，取值如下：

|  |  |  |
| --- | --- | --- |
| 名称 | 类型 | 说明 |
| status | 对象 | 必需，状态 |
| timeStampToken | 字符串 | 可选，时间戳Token的Hex编码 |

status的取值如下：

|  |  |  |
| --- | --- | --- |
| 名称 | 类型 | 说明 |
| status | 对象 | 必需，状态 |
| statusString | 字符串数组 | 可选，每个元素是一个说明 |
| failInfo | 对象 | 可选，错误原因 |

上表的status的取值如下：

|  |  |  |
| --- | --- | --- |
| 名称 | 类型 | 说明 |
| encodeType | 字符串 | 必需，int或者name |
| value | 整数或者字符串 | 必需，状态值或者状态的字符串 |

failInfo的取值如下：

|  |  |  |
| --- | --- | --- |
| 名称 | 类型 | 说明 |
| encodeType | 字符串 | 必需，bit或者name |
| value | 布尔值数组或者字符串数组 | 必需，每个元素是按位的布尔值或者错误名称 |