```
    revocationTime              GeneralizedTime,
    revocationReason    [0]     EXPLICIT CRLReason OPTIONAL }
UnknownInfo ::= NULL -- this can be replaced with an enumeration
```

16.1.2 SOCSP

由于 OCSP 请求包和响应包需要数字签名，从而导致其执行效率不高。为提高 OCSP 的响应速度，采用 MAC 算法代替数字签名，形成简化版的 OCSP，称作 SOCSP（Simple Online Certificate Status Protocol）。

1. SOCSP 请求包

SOCSP 请求包格式用 ASN.1 描述如下：

```
OCSPThinRequest     ::= SEQUENCE {
    tbsThinRequest   TBSThinRequest,        --请求信息
    mac                 MacData             --请求信息的 MAC 值
}
TBSThinRequest ::= SEQUENCE {
    version     [0] EXPLICIT Version DEFAULT   v1,     --版本号
    random      BIT STRING,                            --随机数
    serialList          SEQUENCE OF CertificateSerialNumber   --请求查询证书的序列号
}
MacData ::= SEQUENCE {
    mac                 DigestInfo,
    macSalt             OCTET STRING,
    iterations          INTEGER DEFAULT   1     --默认为 1, HASH 的反复次数
}
DigestInfo ::= SEQUENCE {
    digestAlgorithm             DigestAlgorithmIdentifier,
    digest                      Digest
}
Digest ::= OCTET STRING
```

与 OCSP 不同，SOCSP 的请求数据需要进行基于对称密钥的 MAC 计算而不是基于公私钥对的数字签名。为了提高系统的效率，SOCSP 没有设置扩展，所以用随机数来防止重放攻击，随机数以 random 字段出现在请求数据中，random 的长度≥128 位。

MAC 计算是基于 TBSThinRequest 结构的 DER 编码，加上共享密钥与 macSalt 的模 2 加进行的，macSalt 保证相同数据 MAC 的结果不相同。MAC 算法由 digestAlgorithm 标识，支持符合国家规定的 HMAC 算法。

2. SOCSP 响应包

SOCSP 响应包格式用 ASN.1 描述如下：

```
OCSPThinResponse ::=SEQUENCE {                          --响应数据
    tbsThinResponse         TBSThinResponse,            --响应信息
    Mac                         MacData                 --响应信息的 MAC 值
```

```
}
TBSThinResponse      ::= SEQUENCE {
    version                  [0] EXPLICIT Version DEFAULT v1,          --版本号
    random                   BIT STRING,                               --随机数
    thinResponseStatus           SEQUENCE OF OCSPThinResponseStatus  --响应状态
}
OCSPThinResponseStatus ::= ENUMERATED {
    successful          (0),  --Response has valid confirmations   --响应被有效确认
    malformedRequest    (1),  --Illegal confirmation request       --不完整的请求
    internalError       (2),  --Internal error in issuer           --内部错误
    unauthorized        (3),  --Request unauthorized               --未授权
}
```

从响应数据定义来看，响应包含了两个信息：响应信息和对响应信息进行 MAC 计算的值。响应数据包含：版本号、随机数、响应状态。随机数 random 与请求中的 random 相同；响应状态有 4 种：响应被有效确认、不完整的请求、内部错误、未授权。MacData 数据定义与请求中一致。

16.2　CRL 服务

CRL 是证书作废列表，也称作黑名单。CA 系统通过 CRL 机制定期对外发布已作废（或已冻结）的证书序列号列表。用户端将最新的 CRL 下载到本地，通过解析 CRL 就可获得已作废证书的序列号、作废原因及作废时间等。

CRL 包含每个已作废证书的序列号、作废原因、作废时间等信息，但不含证书的具体内容。CA 系统定期生成新的 CRL，并将已过期证书从 CRL 中删除。

CRL 可以发布到 LDAP 中，也可以文件形式发布到网站上。

IETF RFC 3280 规定了 X.509 CRL 的格式，包括基本域组成、CRL 内容和扩展项。

16.2.1　基本域组成（CertificateList）

CRL 由 3 个域组成，具体见表 16-1。

表 16-1　CRL 基本域组成

分类	标识	说明
CRL 内容（待签名）	tbsCertList	包含签发者信息、签发时间、已作废证书等
签名算法	signatureAlgorithm	包括摘要算法和公钥算法，如 sha1WithRSAEncryption，由算法标识和算法参数组成
签名值	signatureValue	使用签名算法，对 CRL 内容 tbsCertList 进行签名后的结果

CRL 基本域格式用 ASN.1 描述如下：

```
CertificateList  ::=  SEQUENCE  {
    tbsCertList              TBSCertList,
    signatureAlgorithm       AlgorithmIdentifier,
```

```
    signatureValue          BIT STRING    }
AlgorithmIdentifier    ::=  SEQUENCE    {
    Algorithm               OBJECT IDENTIFIER,
    Parameters              ANY DEFINED BY algorithm OPTIONAL    }
```

16.2.2 CRL 内容（tbsCertList）

CRL 内容见表 16-2。

表 16-2 CRL 内容

分类	标识	说明
版本号	version	用于区分证书格式版本，最新版本为 v2
签名算法	signature	必须与基本域中的签名算法相同
CRL 签发者	issuer	用于区分 CRL 签发者，包含 CRL 签发者身份信息
本次签发时间	thisUpdate	本次 CRL 签发时间
下次签发时间	nextUpdate	下次 CRL 签发时间
作废证书集	revokedCertificates	包含已作废证书相关信息
扩展项	crlExtensions	包含其他可扩展信息

CRL 内容格式用 ASN.1 描述如下：

```
TBSCertList  ::=  SEQUENCE  {
    version         Version OPTIONAL,       -- if present, MUST be v2
    signature           AlgorithmIdentifier,
    issuer              Name,
    thisUpdate          Time,
    nextUpdate          Time OPTIONAL,
    revokedCertificates   SEQUENCE   OF   RevokedCertificate OPTIONAL,
    crlExtensions           [0]   EXPLICIT Extensions OPTIONAL
                                  -- if present, MUST be v2
}
```

1. 版本号 version

version 用于区分 CRL 格式版本，最新版本为 v2。当使用扩展项 crlExtensions 时，version=v2。当 revokedCertificates 使用 crlEntryExtensions 属性时，version=v2。

version 格式用 ASN.1 描述如下：

```
Version  ::=  INTEGER  {  v1(0), v2(1) }
```

2. 签名算法 signature

signature 必须与基本域中签名算法相同，即：signature= CertificateList→signatureAlgorithm。

signature 格式用 ASN.1 描述如下：

```
AlgorithmIdentifier  ::=  SEQUENCE  {
    Algorithm               OBJECT IDENTIFIER,
    Parameters              ANY DEFINED BY algorithm OPTIONAL    }
```

3. CRL 签发者 issuer

issuer 用于区分 CRL 签发者，必须包含一个 X.500 DN 项。DN 是 Distinguished Name 的缩写，表示可识别的名称，且 DN 项被定义为 X.501 规范中的 Name 类型。

CRL 签发者编码规则与证书签发者相同（具体参见“第 9 章”）。

issuer 格式用 ASN.1 描述如下：

```
Name ::= CHOICE {     RDNSequence     }
RDNSequence ::= SEQUENCE OF RelativeDistinguishedName
RelativeDistinguishedName ::=    SET OF AttributeTypeAndValue
AttributeTypeAndValue ::= SEQUENCE {
    type        AttributeType,
    value       AttributeValue }
AttributeType ::= OBJECT IDENTIFIER
AttributeValue ::= ANY DEFINED BY AttributeType
```

4. 本次签发时间 thisUpdate 和下次签发时间 nextUpdate

thisUpdate 表示当前 CRL 的签发时间。

nextUpdate 表示下次 CRL 的最晚签发时间，即：下次 CRL 的实际签发时间可以早于 nextUpdate，但不能晚于 nextUpdate。同时，nextUpdate 不能早于已有 CRL 的签发时间。

thisUpdate 和 nextUpdate 均可以采用两种格式：UTCTime 和 GeneralizedTime。UTCTime 用 2 位数字表示年份，GeneralizedTime 用 4 位数字表示年份。2049 年以前的时间可采用 UTCTime 格式，但 2050 年及以后的时间必须采用 GeneralizedTime 格式。当采用 UTCTime 格式时，如 2 位年份数字 YY 小于 50 时，则年份应该解释为 20YY 年；当 YY 大于或等于 50 时，年份应该解释为 19YY 年。

5. 作废证书集 revokedCertificates

revokedCertificates 包含所有已作废证书的相关信息。当没有作废证书时，该字段不应出现。

revokedCertificates 格式用 ASN.1 描述如下：

```
revokedCertificates     SEQUENCE    OF    RevokedCertificate OPTIONAL,
RevokedCertificate    ::=     SEQUENCE    {
    userCertificate         CertificateSerialNumber,
    revocationDate          Time,
    crlEntryExtensions      Extensions OPTIONAL
                             -- if present, MUST be v2
    }
CertificateSerialNumber    ::=    INTEGER
```

其中，revokedCertificates 由多个 CRL 条目组成。每个 CRL 条目包含单个作废证书的信息，其格式定义为 RevokedCertificate 类型，包括证书序列号 userCertificate、作废时间 revocationDate 和 CRL 条目扩展项 crlEntryExtensions。

6. 扩展项 crlExtensions 和 crlEntryExtensions

crlExtensions 用于 CRL 信息扩展，可包含多项扩展信息。

crlEntryExtensions 用于 CRL 条目信息扩展，可包含多项扩展信息。

crlExtensions 和 crlEntryExtensions 格式用 ASN.1 描述如下：

```
Extensions  ::=  SEQUENCE SIZE (1..MAX) OF Extension
Extension   ::=  SEQUENCE  {
     extnID        OBJECT IDENTIFIER,
     critical      BOOLEAN DEFAULT FALSE,
     extnValue     OCTET STRING   }
```

crlExtensions 和 crlEntryExtensions 只能出现在版本 v2 格式中。

每个扩展项都可设置为关键项（critical=TRUE）或非关键项（critical=FALSE）。如果遇到未知的关键扩展项，则必须拒绝该 CRL；如果遇到未知的非关键扩展项，可以忽略该扩展项。

每个扩展项由一个 OID 和一个 ASN.1 结构组成。OID 赋值给 extnID，ASN.1 编码后的结构赋值给 extnValue。

16.2.3 CRL 扩展项 crlExtensions

CRL 扩展项见表 16-3。

表 16-3 CRL 扩展项

/	扩展项	OID	是否关键项	说明
1	AuthorityKeyIdentifier	id-ce 35	FALSE	CRL 签发者密钥标识
2	IssuerAltName	id-ce 18	FALSE	CRL 签发者别名
3	CRLNumber	id-ce 20	FALSE	CRL 编号
4	DeltaCRLIndicator	id-ce 27	TRUE	增量 CRL 指示器
5	IssuingDistributionPoint	id-ce 28	TRUE	CRL 发布点
6	FreshestCRL	id-ce 46	FALSE	最新 CRL 或增量 CRL

注：id-ce OBJECT IDENTIFIER ::= { joint-iso-ccitt(2) ds(5) 29 }

1. authorityKeyIdentifier

authorityKeyIdentifier 扩展项用于区分 CRL 签发者的公钥。当 CRL 签发者拥有多个公私钥对用于签发 CRL 时，必须使用该扩展项。

该扩展项必须设置为非关键项（critical=FALSE）。

authorityKeyIdentifier 格式用 ASN.1 描述如下：

```
id-ce-authorityKeyIdentifier OBJECT IDENTIFIER ::=  { id-ce 35 }
AuthorityKeyIdentifier ::= SEQUENCE {
      keyIdentifier             [0] KeyIdentifier            OPTIONAL,
      authorityCertIssuer       [1] GeneralNames             OPTIONAL,
      authorityCertSerialNumber [2] CertificateSerialNumber OPTIONAL   }
KeyIdentifier ::= OCTET STRING
```

authorityKeyIdentifier 基于 CRL 签发者证书中的内容生成，主要有两种生成方式：基于 subjectKeyIdentifier、基于 issuer 和 serialNumber。当基于 subjectKeyIdentifier 生成时，