keyIdentifier 通常等于 CRL 签发者证书中的 subjectKeyIdentifier。

2．issuerAltName

issuerAltName 扩展项表示 CRL 签发者的别名，可包含多个。别名形式包括电子邮箱、DNS 名称、IP 地址、URI 等，其中 DNS 名称也可以使用 issuer 中的 DN 项 domainComponent 表示。

该扩展项必须设置为非关键项（critical=FALSE）。

issuerAltName 格式用 ASN.1 描述如下：

```
id-ce-issuerAltName OBJECT IDENTIFIER ::=   { id-ce 18 }
IssuerAltName ::= GeneralNames
GeneralNames ::= SEQUENCE SIZE (1..MAX) OF GeneralName
GeneralName ::= CHOICE {
    otherName                   [0]     OtherName,
    rfc822Name                  [1]     IA5String,
    dNSName                     [2]     IA5String,
    x400Address                 [3]     ORAddress,
    directoryName               [4]     Name,
    ediPartyName                [5]     EDIPartyName,
    uniformResourceIdentifier   [6]     IA5String,
    iPAddress                   [7]     OCTET STRING,
    registeredID                [8]     OBJECT IDENTIFIER }
```

3．cRLNumber

cRLNumber 扩展项表示当前 CRL 的编号，采用顺序递增的整数表示（monotonically increasing sequence number），用于快速区分不同的 CRL。cRLNumber 同时支持全量 CRL（包含所有作废证书）和增量 CRL（只包含新作废证书）。

该扩展项必须设置为非关键项（critical=FALSE）。

cRLNumber 格式用 ASN.1 描述如下：

```
id-ce-cRLNumber OBJECT IDENTIFIER ::= { id-ce 20 }
CRLNumber ::= INTEGER (0..MAX)
```

如果 CRL 签发者既生成全量 CRL，也生成增量 CRL，则这两类 CRL 必须统一编号（share one numbering sequence），即不允许独立编号。如果一个全量 CRL 和一个增量 CRL 同时生成，则它们的 cRLNumber 必须相同，且必须包含相同的作废证书集，即该增量 CRL 与上一个全量 CRL 合并后，与该全量 CRL 必须包含相同的作废证书集。

如果 CRL 签发者在不同的时间生成 2 个 CRL（2 个全量 CRL，2 个增量 CRL，或者 1 个全量 CRL 和 1 个增量 CRL），则这 2 个 CRL 必须使用不同的 cRLNumber。也就是说，如果两个 CRL 中的 thisUpdate 不相同，则其 cRLNumber 也不能相同。

4．deltaCRLIndicator

deltaCRLIndicator 扩展项表示增量 CRL 指示器，若存在，则说明当前 CRL 是增量 CRL。增量 CRL 只包含最新的作废证书（上次发布 CRL 后），并不包含所有的作废证书。在有些环境下使用增量 CRL，可以显著降低网络负载和处理时间。增量 CRL 通常比全量 CRL 要小，因此应用系统获取增量 CRL 将比获取全量 CRL 消耗更少的网络带宽。本地保存作废

证书集时，建议不使用 CRL 结构而采用其他格式，因为获取增量 CRL 后，可以将其包含的新作废证书很方便地增加到本地保存的已有作废证书集中。

对于增量 CRL，必须包含该扩展项，且该扩展项必须设置为关键项（critical=TRUE）。

cRLNumber 格式用 ASN.1 描述如下：

```
id-ce-deltaCRLIndicator OBJECT IDENTIFIER ::= { id-ce 27 }
BaseCRLNumber ::= CRLNumber
```

其中，BaseCRLNumber 表示全量 CRL 的编号，该全量 CRL 是当前增量 CRL 的起点。也就是说，当前增量 CRL 与该全量 CRL 合并后，则包含所有的作废证书。

一个全量 CRL 和一个增量 CRL 允许合并，必须满足以下 4 个条件：

① 全量 CRL 和增量 CRL 具有相同的 CRL 签发者。

② 全量 CRL 和增量 CRL 具有相同的证书范围（scope）。只要满足其中一个条件，就可以认为两个 CRL 具有相同的证书范围。条件一：两个 CRL 均不包含 issuingDistributionPoint 扩展项；条件二：两个 CRL 的 issuingDistributionPoint 扩展项内容完全相同。

③ 全量 CRL 的 CRLNumber 等于或大于增量 CRL 的 BaseCRLNumber。也就是说，该全量 CRL 包含编号为 BaseCRLNumber 的全量 CRL 中所有作废证书。

④ 全量 CRL 的 CRLNumber 小于增量 CRL 的 CRLNumber。

5. issuingDistributionPoint

issuingDistributionPoint 扩展项表示 CRL 发布点和证书范围。通过该扩展项可显示出当前 CRL 的证书覆盖范围，如只覆盖终端证书、CA 证书、属性证书，或只覆盖特定作废原因的证书。CRL 由 CRL 签发者的私钥进行签名，但 CRL 发布点不需要自己的公私钥对。如果 CRL 存储在 X.500 目录服务器中，则应该保存到“CRL 发布点”对应的目录条目中，不应保存到“CRL 签发者”对应的目录条目中。

该扩展项必须设置为关键项（critical=TRUE）。

issuingDistributionPoint 格式用 ASN.1 描述如下：

```
id-ce-issuingDistributionPoint OBJECT IDENTIFIER ::= { id-ce 28 }
issuingDistributionPoint ::= SEQUENCE {
    distributionPoint          [0] DistributionPointName OPTIONAL,
    onlyContainsUserCerts      [1] BOOLEAN DEFAULT FALSE,
    onlyContainsCACerts        [2] BOOLEAN DEFAULT FALSE,
    onlySomeReasons            [3] ReasonFlags OPTIONAL,
    indirectCRL                [4] BOOLEAN DEFAULT FALSE,
    onlyContainsAttributeCerts [5] BOOLEAN DEFAULT FALSE }
```

其中，作废原因必须包含在 onlySomeReasons 字段中。如果 onlySomeReasons 没有出现，CRL 发布点必须包含所有作废原因的作废信息。例如，作废原因为 keyCompromise（1）、cACompromise（2）、aACompromise（8）的作废证书可以在 A 发布点，其他作废证书可以在 B 发布点。

如果 distributionPoint 存在且包含一个 URI，则该 URI 必须链接到最新的 CRL，且只能使用绝对地址，必须指定主机名称，访问方式可以是 ftp、http、mail 和 ldap。

如果 CRL 签发者与证书签发者不相同，则 indirectCRL 必须设置为 TRUE；应在 CRL 条目扩展项 CertificateIssuer 中表明证书签发者。

6．freshestCRL（Delta CRL Distribution Point）

freshestCRL 扩展项用于确定如何获取当前全量 CRL 对应的增量 CRL 信息。该扩展项不能出现在增量 CRL 中。

该扩展项应该设置为非关键项（critical=FALSE）。

freshestCRL 格式用 ASN.1 描述如下：

```
id-ce-freshestCRL OBJECT IDENTIFIER ::=   { id-ce 46 }
FreshestCRL ::= CRLDistributionPoints
CRLDistributionPoints ::= SEQUENCE SIZE (1..MAX) OF DistributionPoint
DistributionPoint ::= SEQUENCE {
    distributionPoint          [0]     DistributionPointName OPTIONAL,
    reasons                    [1]     ReasonFlags OPTIONAL,
    cRLIssuer                  [2]     GeneralNames OPTIONAL }
DistributionPointName ::= CHOICE {
    fullName                       [0]     GeneralNames,
    nameRelativeToCRLIssuer    [1]     RelativeDistinguishedName }
```

其中，只有 distributionPoint 有效；reasons 和 cRLIssuer 无效，应忽略。

16.2.4　CRL 条目扩展项 crlEntryExtensions

CRL 条目扩展项见表 16-4。

表 16-4　CRL 条目扩展项

/	扩展项	OID	是否关键项	说明
1	ReasonCode	id-ce 21	FALSE	作废原因代码
2	HoldInstructionCode	id-ce 23	FALSE	冻结指令代码
3	InvalidityDate	id-ce 24	FALSE	证书无效日期
4	CertificateIssuer	id-ce 29	TRUE	证书签发者

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

1．reasonCode

reasonCode 扩展项表示证书作废原因。

该扩展项应该设置为非关键项（critical=FALSE）。

reasonCode 格式用 ASN.1 描述如下：

```
id-ce-cRLReason OBJECT IDENTIFIER ::= { id-ce 21 }
reasonCode ::= { CRLReason }
CRLReason ::= ENUMERATED {
    unspecified                (0),
    keyCompromise              (1),     --表示密钥泄露
    cACompromise               (2),     --表示 CA 泄露
```

```
affiliationChanged      (3),    --表示关系变更
superseded              (4),    --表示废弃
cessationOfOperation    (5),    --表示操作中止
certificateHold         (6),    --表示证书冻结
removeFromCRL           (8),    --表示从 CRL 删除
privilegeWithdrawn      (9),    --表示权限撤销
aACompromise            (10)    --表示 AA 泄露
}
```

2. holdInstructionCode

holdInstructionCode 扩展项表示冻结指令代码。当碰到作废原因为 certificateHold 的证书时，应按照该代码采取相应的行动。

该扩展项应该设置为非关键项（critical=FALSE）。

holdInstructionCode 格式用 ASN.1 描述如下：

```
id-ce-holdInstructionCode OBJECT IDENTIFIER ::= { id-ce 23 }
holdInstructionCode ::= OBJECT IDENTIFIER
```

常用的指令代码用 ASN.1 描述如下：

```
holdInstruction      OBJECT IDENTIFIER ::=
                { iso(1) member-body(2) us(840) x9-57(10040) 2 }
id-holdinstruction-none     OBJECT IDENTIFIER ::= {holdInstruction 1}
id-holdinstruction-callissuer    OBJECT IDENTIFIER ::= {holdInstruction 2}
id-holdinstruction-reject OBJECT IDENTIFIER ::= {holdInstruction 3}
```

其中，id-holdinstruction-callissuer 表示联系证书签发者或拒绝该证书。id-holdinstruction-reject 表示拒绝该证书。id-holdinstruction-none 表示不采取任何行动，与不包含该扩展项表示相同含义。

3. invalidityDate

invalidityDate 扩展项表示知道或怀疑私钥泄露或证书无效的时间。该时间可能早于 CRL 条目的作废时间。

该扩展项应该设置为非关键项（critical=FALSE）。

invalidityDate 格式用 ASN.1 描述如下：

```
id-ce-invalidityDate OBJECT IDENTIFIER ::= { id-ce 24 }
invalidityDate ::=   GeneralizedTime
```

4. certificateIssuer

certificateIssuer 扩展项表示证书签发者。如果该 CRL 条目扩展项存在，则 CRL 扩展项 issuingDistributionPoint 的 indirectCRL 必须设置为 TRUE。如果第一个 CRL 条目的 certificateIssuer 扩展项不存在，则表示证书签发者与 CRL 签发者相同。如果某个 CRL 条目的 certificateIssuer 扩展项不存在，则表示该 CRL 条目的证书签发者与上一个 CRL 条目相同。

该扩展项应该设置为关键项（critical=TRUE）。

certificateIssuer 格式用 ASN.1 描述如下：

```
id-ce-certificateIssuer     OBJECT IDENTIFIER ::= { id-ce 29 }
certificateIssuer ::=       GeneralNames
```

16.3　LDAP 服务

LDAP 是 Lightweight Directory Access Protocol（轻型目录访问协议）的缩写。CA 系统通过 LDAP 机制对外发布所有证书及 CRL。用户端可以通过 LDAP 协议访问 LDAP 服务器，按需下载满足条件的证书和 CRL。

16.3.1　发布数字证书到 LDAP

1．定义 LDAP schema

LDAP schema 是一个规则集，定义了 LDAP 目录所应遵循的结构和规则，它决定哪些信息可以存放在目录服务中，以及在客户端和目录服务器查询交互中如何处理信息。目录服务器在存储新数据或修改现有数据时，会检查数据是否满足 schema 规则。当客户端或服务器比较两个属性值时，它们会使用 schema 规定的比较算法。

LDAP schema 主要由四个元素组成：对象类（objectClass）、属性（attribute）、语法（Syntax）、匹配规则（Matching Rules）。

LDAP schema 定义后，将形成 LDAP 目录树结构。例如，针对 DN 项为“c=?,st=?,o=?,ou=?,cn=?”，定义 LDAP schema 后形成的 LDAP 目录树如图 16-1 所示。

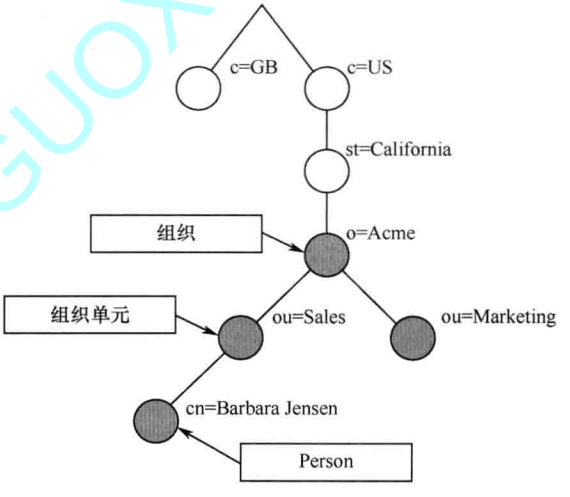

图 16-1　LDAP 目录树（传统命名方式）

2．添加数字证书条目到 LDAP

CA 系统签发每个数字证书后，将实时或定期通过 API 方式将该数字证书发布到 LDAP 服务器。

CA 系统添加数字证书条目到 LDAP 的步骤主要包括：

① 打开 LDAP Server 连接。ldap_open()返回连接句柄，允许多个连接同时打开。

② 同 LDAP Server 进行身份认证。ldap_bind()及相关函数支持多种不同的认证方法。