国 际 电 信 联 盟

ITU-T

国际电信联盟
电信标准化部门

X.509

(08/2005)

X系列：数据网、开放系统通信和安全性

号码簿

信息技术 — 开放系统互连 — 号码簿: 公开密钥和属性证书框架

ITU-T X.509建议书

国际电信联盟

ITU-T X系列建议书

数据网、开放系统通信和安全性

公众数据网	
业务和设施	X.1-X.19
接口	X.20-X.49
传输、信令和交换	X.50-X.89
网络概貌	X.90-X.149
维护	X.150-X.179
管理安排	X.180-X.199
开放系统互连	
模型和记法	X.200-X.209
服务限定	X.210-X.219
连接式协议规范	X.220-X.229
无连接式协议规范	X.230-X.239
PICS书写形式	X.240-X.259
协议标识	X.260-X.269
安全协议	X.270-X.279
层管理对象	X.280-X.289
一致性测试	X.290-X.299
网间互通	
概述	X.300-X.349
卫星数据传输系统	X.350-X.369
以IP为基础的网络	X.370-X.379
报文处理系统	X.400-X.499
号码簿	**X.500-X.599**
OSI组网和系统概貌	
组网	X.600-X.629
效率	X.630-X.639
服务质量	X.640-X.649
命名、寻址和登记	X.650-X.679
抽象句法记法1(ASN.1)	X.680-X.699
OSI管理	
系统管理框架和结构	X.700-X.709
管理通信服务和协议	X.710-X.719
管理信息的结构	X.720-X.729
管理功能和ODMA功能	X.730-X.799
安全	X.800-X.849
OSI应用	
托付、并发和恢复	X.850-X.859
事务处理	X.860-X.879
远程操作	X.880-X.889
ASN.1的一般应用	X.890-X.899
开放分布式处理	X.900-X.999
电信安全	X.1000-

欲了解更详细信息，请查阅*ITU-T*建议书目录。

国际标准 ISO/IEC 9594-8
ITU-T X.509建议书

信息技术 — 开放系统互连 — 号码簿：公开密钥和属性证书框架

摘　要

本建议书 | 国际标准为公开密钥证书和属性证书定义了一个框架。其他标准机构可以使用这些框架来勾画其有关公开密钥基础设施（PKI）和特权管理基础设施（PMI）的应用。另外，本建议书 | 国际标准为号码簿向其用户提供鉴权服务定义了一个框架。它描述了两个级别的鉴权：简单鉴权，利用口令来确认提出要求的实体；强鉴权，涉及利用加密技术形成的证书。简单鉴权对未经授权的访问提供了一些有限的保护，只有强鉴权才能作为提供安全服务的基础。

来　源

ITU-T 第 17 研究组（2005-2008）按照 ITU-T A.8 建议书规定的程序，于 2005 年 8 月 29 日批准了 ITU-T X.509 建议书。同一文本还以 ISO/IEC 9594-8 的形式发布。

前　言

国际电信联盟（ITU）是从事电信领域工作的联合国专门机构。ITU-T（国际电信联盟电信标准化部门）是国际电信联盟的常设机构，负责研究技术、操作和资费问题，并且为在世界范围内实现电信标准化，发表有关上述研究项目的建议书。

每四年一届的世界电信标准化全会（WTSA）确定 ITU-T 各研究组的研究课题，再由各研究组制定有关这些课题的建议书。

WTSA 第 1 号决议规定了批准建议书须遵循的程序。

属 ITU-T 研究范围的某些信息技术领域的必要标准，是与国际标准化组织（ISO）和国际电工技术委员会（IEC）合作制定的。

注

本建议书为简明扼要起见而使用的“主管部门”一词，既指电信主管部门，又指经认可的运营机构。

遵守本建议书的规定是以自愿为基础的，但建议书可能包含某些强制性节（以确保例如互操作性或适用性等），只有满足所有强制性节的规定，才能达到遵守建议书的目的。“应该”或“必须”等其他一些强制性用语及其否定形式被用于表达特定要求。使用此类用语不表示要求任何一方遵守本建议书。

知识产权

国际电联提请注意：本建议书的应用或实施可能涉及使用已申报的知识产权。国际电联对无论是其成员还是建议书制定程序之外的其他机构提出的有关已申报的知识产权的证据、有效性或适用性不表示意见。

至本建议书批准之日止，国际电联已经收到实施本建议书可能需要的受专利保护的知识产权的通知。但需要提醒实施者注意的是，这可能并非最新信息，因此特大力提倡他们通过下列网址查询电信标准化局（TSB）的专利数据库：http://www.itu.int/ITU-T/ipr/。

目 录

引言

本建议书 | 国际标准连同本系列其他建议书 | 国际标准是为方便信息处理系统之间的互连以提供号码簿服务而制定的。所有这些系统的集合，连同它们所拥有的号码簿信息可被视为一个整体，被称为*号码簿*。号码簿所拥有的信息，总称为号码簿信息库(DIB)，典型地被用于推动对象之间的通信、与对象的通信或有关对象的通信等，这些对象如应用实体、个人、终端和分发表等。

号码簿在开放系统互连中扮演了重要角色，其目标是在它们自身的互连标准之外做最少的技术约定的情况下，允许以下各种信息处理系统之间的互连：

— 来自不同生产厂商；

— 具有不同的管理；

— 具有不同的复杂程度；以及

— 具有不同的年代。

许多应用都有安全方面的需求，以便抵御对信息通信的威胁。实际上，所有的安全服务都有赖于可靠地知晓各通信方的身份，即鉴权。

本建议书 | 国际标准为公开密钥证书定义了一个框架。该框架包括用于代表证书自身以及有关不再可信的已发放证书撤消通知的数据对象规范。本规范定义了公开密钥证书框架，同时它还定义了公开密钥基础设施（PKI）的某些关键部件，但它并不完整定义一个 PKI。不过，本规范为完整的 PKI 以及将要建立的规范奠定了基础。

同样，本建议书 | 国际标准为属性证书定义了一个框架。该框架包括用于代表证书自身以及有关不再可信的已发放证书撤消通知的数据对象规范。本规范定义了属性证书框架，同时它还定义了特权管理基础设施（PMI）的某些关键部件，但它并不完整定义一个 PMI。不过，本规范为完整的 PMI 以及将要建立的规范奠定了基础。

还定义了用于持有号码簿中 PKI 和 PMI 对象的信息对象以及用于比较出现值和保存值的信息对象。

本建议书 | 国际标准还定义了一个用于号码簿向其用户提供鉴权服务的框架。

本建议书|国际标准提供了一个基础框架，在此框架基础上，其他标准化组织和业界论坛可以定义工业配置集。在本框架中定义为可选的许多特性，可通过配置集的说明，在某种环境下作为必选特性来使用。目前本建议书|国际标准的第 5 版是原有第 4 版的修订和增强，但不是替代。在系统实现时仍可以声明为遵循第 4 版。然而，在某些方面，将不再支持第 4 版（即不再消除一些报告上来的错误）。建议在系统实现时尽快遵循第 5 版。

这是第 5 版本，规定了第 1 版本、第 2 版本和第 3 版本的公开密钥证书，以及第 1 版本和第 2 版本的证书撤消清单。该版本还规定了第 2 版本的属性证书。

在一个带第 3 版本公开密钥证书和第 2 版本证书撤消清单的早期版本中，增加了延展性功能，并从其一开始就溶入到了属性证书中。该功能在第 7 节中规定。预计对该版本的任何增强都可以利用本功能提供，而无需创建新的版本。

附件 A 是本建议书 | 国际标准的组成部分，提供了 ASN.1 模块，它包含与各框架有关的所有定义。

附件 B 是本建议书 | 国际标准的组成部分，提供了用于产生和处理证书撤消清单的规则。

附件 C 不是本建议书 | 国际标准的组成部分，提供了 delta-CRL 发布的例子。

附件 D 不是本建议书 | 国际标准的组成部分，提供了特权策略语法和特权属性的例子。

附件 E 不是本建议书 | 国际标准的组成部分，是对公开密钥密码系统的一个概述。

附件 F 是本建议书 | 国际标准的组成部分，定义了在没有正式寄存器的情况下指派给鉴权和加密算法的对象标识符。

附件 G 不是本建议书 | 国际标准的组成部分，包含认证通路约束使用的例子。

附件 H 不是本建议书 | 国际标准的组成部分，在认证通路验证过程中，在处理证书策略时，为具有 PKI 功能的应用提供指南。

附件 I 不是本建议书 | 国际标准的组成部分，为 keyUsage 证书扩展中 contentCommitment 位的使用提供指南。

附件 J 不是本建议书 | 国际标准的组成部分，包含一个按字母顺序的本规范中信息项目定义清单。

附件 K 不是本建议书 | 国际标准的组成部分，列出了各修正和缺陷报告，纳入它们，形成本建议书 | 国际标准的这个版本。

国际标准
ITU-T 建议书

信息技术 — 开放系统互连 — 号码簿：公开密钥和属性证书框架

第1部分 — 概述

1 范围

通过提供一系列框架，本建议书 | 国际标准阐述了鉴权领域的某些安全需求以及其他安全服务，这些框架是完整服务的基础。尤其是，本建议书 | 国际标准为以下内容定义了框架：

— 公开密钥证书；

— 属性证书；

— 鉴权服务。

在本建议书 | 国际标准中定义的公开密钥证书框架包括有关公开密钥基础设施（PKI）的信息对象定义，包括公开密钥证书以及证书撤消清单（CRL）。属性证书框架包括有关特权管理基础设施（PMI）的信息对象定义，包括属性证书以及证书撤消清单（ACRL）。本规范还提供了有关发放、管理、使用和撤消证书的框架。对两种证书类型和所有撤消清单方案，在所定义的格式中包括了一个延展性机制。本建议书 | 国际标准还包括一系列标准扩展，它们有望通用于众多 PKI 和 PMI 应用。在本建议书 | 国际标准包括了方案部件，包括用于保存号码簿中 PKI 和 PMI 对象的对象类别、属性类型和匹配规则。这些框架之外的其他 PKI 和 PMI 元素，如密钥和证书管理协议、操作协议、额外的证书和 CRL 扩展等，有望由其他标准实体（如 ISO TC68、IETF 等）来定义。

在本建议书 | 国际标准中定义的鉴权方案是通用的，可以应用于众多应用和环境。

号码簿利用公开密钥证书和属性证书，在本建议书 | 国际标准中还定义了有关号码簿使用这些工具的框架。号码簿利用公开密钥技术，包括证书，实现强鉴权、经签名与/或加密的操作，以及保存号码簿中的、经签名与/或加密的数据。号码簿可以利用属性证书来实现基于规则的访问控制。虽然在本规范中提供了有关这些内容的框架，但在号码簿规范的完整集中提供有关号码簿使用这些框架的完整定义以及号码簿提供的相关服务及其部件。

在鉴权服务框架中，本建议书 | 国际标准还：

— 规定了号码簿所持有的鉴权信息的形式；

— 描述了如何从号码簿获取鉴权信息；

— 描述了所做的、有关鉴权信息如何形成以及如何置于号码簿中的假设；

— 定义了三种应用使用该鉴权信息实施鉴权的方式，并描述了鉴权如何支持其他安全服务。

本建议书 | 国际标准描述了两个级别的鉴权：简单鉴权，利用口令来确认所要求的实体；强鉴权，涉及利用加密技术形成的证书。简单鉴权对未经授权的访问提供了一些有限的保护，只有强鉴权才能作为提供安全服务的基础。它无意将之建成为一个通用的鉴权框架，但它可以通用于各应用，认为这些技术是适当的。

鉴权（和其他安全服务）只能在所定义的安全策略范畴内提供。由应用用户来定义其自身的安全策略，它可以由某个标准提供的服务来约束。

由使用鉴权框架的标准定义应用来规定所需实施的协议交换，以便实现基于获自号码簿的鉴权信息的鉴权。应用于从号码簿获得证书的协议为号码簿访问协议（DAP），它在 ITU-T X.519 建议书 | ISO/IEC 9594-5 中规定。

2 规范性参考文献

下列建议书和国际标准所包含的节，在本建议书中的引用而构成本建议书 | 国际标准的节。在出版时，所指出的版本是有效的。所有的建议书和标准都面临修订，使用本建议书 | 国际标准的各方应探讨使用下列建议书和国际标准最新版本的可能性。IEC 和 ISO 的各成员有目前有效的国际标准的目录。国际电联电信标准化局有目前有效的 ITU-T 建议书的清单。

2.1 等同的建议书 | 国际标准

- ITU-T Recommendation X.411 (1999) | ISO/IEC 10021-4:2003, *Information technology – Message Handling Systems (MHS) – Message transfer system: Abstract service definition and procedures*.
- ITU-T Recommendation X.500 (2005) | ISO/IEC 9594-1:2005, *Information technology – Open Systems Interconnection – The Directory: Overview of concepts, models and services*.
- ITU-T Recommendation X.501 (2005) | ISO/IEC 9594-2:2005, *Information technology – Open Systems Interconnection – The Directory: Models*.
- ITU-T Recommendation X.511 (2005) | ISO/IEC 9594-3:2005, *Information technology – Open Systems Interconnection – The Directory: Abstract service definition*.
- ITU-T Recommendation X.518 (2005) | ISO/IEC 9594-4:2005, *Information technology – Open Systems Interconnection – The Directory: Procedures for distributed operation*.
- ITU-T Recommendation X.519 (2005) | ISO/IEC 9594-5:2005, *Information technology – Open Systems Interconnection – The Directory: Protocol specifications*.
- ITU-T Recommendation X.520 (2005) | ISO/IEC 9594-6:2005, *Information technology – Open Systems Interconnection – The Directory: Selected attribute types*.
- ITU-T Recommendation X.521 (2005) | ISO/IEC 9594-7:2005, *Information technology – Open Systems Interconnection – The Directory: Selected object classes*.
- ITU-T Recommendation X.525 (2005) | ISO/IEC 9594-9:2005, *Information technology – Open Systems Interconnection – The Directory: Replication*.
- ITU-T Recommendation X.530 (2005) | ISO/IEC 9594-10:2005, *Information technology – Open Systems Interconnection – The Directory: Use of systems management for administration of the Directory*.
- ITU-T Recommendation X.660 (2004) | ISO/IEC 9834-1:2005, *Information technology – Open Systems Interconnection – Procedures for the operation of OSI Registration Authorities: General procedures, and top arcs of the ASN.1 Object Identifier tree*.
- ITU-T Recommendation X.680 (2002) | ISO/IEC 8824-1:2002, *Information technology – Abstract Syntax Notation One (ASN.1): Specification of basic notation.*
- ITU-T Recommendation X.681 (2002) | ISO/IEC 8824-2:2002, *Information technology – Abstract Syntax Notation One (ASN.1): Information object specification*.
- ITU-T Recommendation X.682 (2002) | ISO/IEC 8824-3:2002, *Information technology – Abstract Syntax Notation One (ASN.1): Constraint specification*.
- ITU-T Recommendation X.683 (2002) | ISO/IEC 8824-4:2002, *Information technology – Abstract Syntax Notation One (ASN.1): Parameterization of ASN.1 specifications*.
- ITU-T Recommendation X.690 (2002) | ISO/IEC 8825-1:2002, *Information technology – ASN.1 encoding rules: Specification of Basic Encoding Rules (BER), Canonical Encoding Rules (CER) and Distinguished Encoding Rules (DER)*.
- ITU-T Recommendation X.691 (2002) | ISO/IEC 8825-2:2002, *Information technology – ASN.1 encoding rules: Specification of Packed Encoding Rules (PER)*.

- ITU-T Recommendation X.812 (1995) | ISO/IEC 10181-3:1996, *Information technology – Open Systems Interconnection – Security frameworks for open systems: Access control framework.*
- ITU-T Recommendation X.813 (1996) | ISO/IEC 10181-4:1997, *Information technology – Open Systems Interconnection – Security frameworks for open systems: Non-repudiation framework*.
- ITU-T Recommendation X.880 (1994) | ISO/IEC 13712-1:1995, *Information technology – Remote Operations: Concepts, model and notation*.
- ITU-T Recommendation X.881 (1994) | ISO/IEC 13712-2:1995, *Information technology – Remote Operations: OSI realizations – Remote Operations Service Element (ROSE) service definition*.

2.2 技术内容相当的配对的建议书 | 国际标准

- CCITT Recommendation X.800 (1991), *Security Architecture for Open Systems Interconnection for CCITT applications*.

  ISO 7498-2:1989, *Information processing systems – Open Systems Interconnection – Basic Reference Model – Part 2: Security Architecture*.

3 定义

就本建议书 | 国际标准而言，采用下列定义。

3.1 OSI 参考模型安全体系结构定义

下列术语在 CCITT X.800 | ISO 7498-2 建议书中规定：

a） 非对称（加密）；
b） 鉴权交换；
c） 鉴权信息；
d） 机密性；
e） 证书；
f） 密码系统；
g） 数据源鉴权；
h） 解密；
i） 数字签名；
j） 加密；
k） 密钥；
l） 口令；
m） 对等实体鉴权；
n） 对称（加密）。

3.2 号码簿模型定义

下列术语在 ITU-T X.501 | ISO/IEC 9594-2 建议书中规定：

a） 属性；
b） 号码簿信息库；
c） 号码簿信息树；
d） 号码簿系统代理；
e） 号码簿用户代理；
f） 不同的名称；
g） 条目；
h） 对象；
i） 根。

3.3 定义

下列术语在本建议书 | 国际标准中规定：

3.3.1 attribute certificate（AC） 属性证书：一种数据结构，由属性机构数字签署，它将某些属性值与其持有者的身份信息进行绑定。

3.3.2 Attribute Authority（AA） 属性机构：一个通过发放属性证书来指派特权的机构。

3.3.3 attribute authority revocation list（AARL） 属性机构撤消清单：一个包含参考文献清单的撤消清单，标明那些发放机构不再认为不再有效的、发放给 AA 的证书。

3.3.4 attribute certificate revocation list（ACRL） 属性证书撤消清单：一个包含参考文献清单的撤消清单，标明那些发放机构不再认为有效的证书。

3.3.5 authentication token;（token）； 鉴权令牌：在强鉴权交换期间传送的信息，它可用于鉴别其发送者。

3.3.6 authority 机构：一个实体，负责发放证书。在本规范中定义了两种类型的机构：负责发放公开密钥证书的认证机构；负责发放属性证书的属性机构。

3.3.7 authority certificate 机构证书：发放给某个机构的证书（例如，发放给一个认证机构或者一个属性机构）。

3.3.8 base CRL 基础 CRL：在产生 dCRL 过程中用作基础的 CRL。

3.3.9 CA-certificate CA-证书：由一个 CA 发放给另一个 CA 的证书。

3.3.10 certificate policy 证书策略：一个经过命名的规则集，用于指明证书对某个特殊团体与/或具有公共安全需求的应用类别的适用性。例如，某个特殊的证书策略可以用于指明某类证书对电子数据交换事务认证的适用性，以便在某个特定的价格范围内销售商品。

3.3.11 certification practice statement（CPS） 认证实施声明：认证机构在发放证书中所用的实施声明。

3.3.12 certificate revocation list（CRL） 证书撤消清单：一个经过签署的清单，用于指明证书发放者不再认为有效的一系列证书。除了通用的术语 CRL 之外，为 CRL 定义了某些特定的 CRL 类型，涵盖了特殊的范围。

3.3.13 certificate user 证书用户：一个实体，它需要确切知道另一个实体的属性与/或公开密钥。

3.3.14 certificate serial number 证书序列号：一个整数值，在发放机构内是惟一的，它明确地关联于该机构发放的某个证书。

3.3.15 certificate-using system 证书使用系统：本号码簿规范中所定义的那些功能的一个实现，供证书的用户使用。

3.3.16 certificate validation 证书验证：确保证书在某个特定时间有效的过程，包括可能构建和处理一个认证通路，并确保在该特定时间该通路上的所有证书是有效的（即不过期或不被撤消）。

3.3.17 certification authority（CA） 认证机构：被一个或多个用户所信任的机构，用于创建和指派公开密钥证书。可选地，认证机构可以创建用户的密钥。

3.3.18 certification authority revocation list（CARL） 认证机构撤消清单：一个包含发放给认证机构的公开密钥证书清单的撤消清单，证书发放者不再认为它有效。

3.3.19 certification path 认证通路：DIT 中一个排好序的、对象公开密钥证书序列，它与通路中最初对象的公开密钥一起，可经处理而获得通路中最终对象的公开密钥。

3.3.20 CRL distribution point CRL 分发点：一个号码簿条目或 CRL 的其他分发源；通过 CRL 分发点分发的 CRL 可以包含只针对某个 CA 发放之证书全集的一个子集的撤消条目，或者可以包含针对多个 CA 的撤消条目。

3.3.21 cross-certificate 交叉证书：一个公开密钥或属性证书，其发放者和对象/持有者与 CA 或 AA 各不相同。CA 和 AA 分别向其他 CA 或 AA 发放交叉证书，作为一种机制来授权对象 CA 的存在（例如，在一个严格的层次结构中），或者认可对象 CA 或持有者 AA 的存在（例如，在一个分布式的信任模型中）。交叉证书结构可以在这两种情况下使用。

批注 [P1]: Page: 4 ... 1
批注 [P2]: Page: 4 ... 2
批注 [P3]: Page: 4 ... 3
批注 [P4]: Page: 4 ... 4
批注 [P5]: Page: 4 ... 5
批注 [P6]: Page: 4 ... 6
批注 [P7]: Page: 4 ... 7
批注 [P8]: Page: 4 ... 8
批注 [P9]: Page: 4 ... 9
批注 [P10]: Page: 4 ... 10
批注 [P11]: Page: 4 ... 11
批注 [P12]: Page: 4 ... 12
批注 [P13]: Page: 4 ... 13
批注 [P14]: Page: 4 ... 14
批注 [P15]: Page: 4 ... 15
批注 [P16]: Page: 4 ... 16
批注 [P17]: Page: 4 ... 17
批注 [P18]: Page: 4 ... 18
批注 [P19]: Page: 4 ... 19
批注 [P20]: Page: 4 ... 20
批注 [P21]: Page: 4 ... 21
批注 [P22]: Page: 4 ... 22
批注 [P23]: Page: 4 ... 23
批注 [P24]: Page: 4 ... 24
批注 [P25]: Page: 4 ... 25
批注 [P26]: Page: 4 ... 26
批注 [P27]: Page: 4 ... 27
批注 [P28]: Page: 4 ... 28
批注 [P29]: Page: 4 ... 29
批注 [P30]: Page: 4 ... 30
批注 [P31]: Page: 4 ... 31
批注 [P32]: Page: 4 ... 32
批注 [P33]: Page: 4 ... 33
批注 [P34]: Page: 4 ... 34
批注 [P35]: Page: 4 ... 35
批注 [P36]: Page: 4 ... 36
批注 [P37]: Page: 4 ... 37
批注 [P38]: Page: 4 ... 38
批注 [P39]: Page: 4 ... 39
批注 [P40]: Page: 4 ... 40
... 41
... 42
... 43

3.3.22 cryptographic system, cryptosystem **加密系统**：从明文到密文的一系列转换，反之亦然，通过密钥选择要用的特殊转换。通常通过数学算法来定义各种转换。

3.3.23 data confidentiality **数据机密性**：可用该服务来为数据提供保护，防止未经授权的透露。鉴权框架支持数据机密性服务。可用它来防止数据截获。

3.3.24 delegation **委托**：将特权从持有该特权的一个实体转让给另一个实体。

3.3.25 delegation path **委托通路**：一个排好序的证书序列，它与特权声明者身份的鉴别一起，可经处理而验证声明者特权的真实性。

3.3.26 delta-CRL（dCRL）：一个本地的撤消清单，它只包含有关以下证书的条目，即其撤消清单的状态在发布所参考的基础 CRL 后发生了变化。

3.3.27 end entity **终端实体**：一个公开密钥证书对象，它使用其专用密钥于签署证书之外的其他目的；或者一个属性证书持有者，它使用其属性来访问某个资源；或者是一个为信任方的实体。

3.3.28 end-entity attribute certificate revocation list（EARL） **终端实体属性证书撤消清单**：一个包含发放给持有者的属性证书的撤消清单，持有者不是 AA，证书发放者不再认为这些证书有效。

3.3.29 end-entity public-key certificate revocation list（EPRL） **终端实体公开密钥证书撤消清单**：一个包含发放给对象的公开密钥证书的撤消清单，持有者不是 CA，证书发放者不再认为这些证书有效。

3.3.30 environmental variables **环境变量**：授权决策所需的那些策略因素，它们不包括在静态结构中，但通过某些本地方式，可使特权验证者使用它们（例如，天数或当前的账目平衡）。

3.3.31 full CRL **完整的 CRL**：一个完整的撤消清单，它包含有关已从特定的范围中撤消的全部证书的条目。

3.3.32 hash function **散列函数**：一个（数学）函数，它从一个大（可能非常大）的范围将值映射至一个较小的范围上。一个“好的”散列函数是这样一个函数，即在将之应用于范围内一个（大的）值的集合时，在整个范围内结果将均匀分布（并显然是随机的）。

3.3.33 holder **持有者**：一个实体，对它已委托某些特权，要么通过机构源直接委托，要么通过另一个属性机构间接委托。

3.3.34 indirect CRL（iCRL） **间接的 CRL**：一个撤消清单，它至少包含有关机构所发放证书的撤消信息，此处所指的机构不是那些发放本 CRL 的机构。

3.3.35 key agreement **密钥协议**：一种在线商定密钥值的方法，甚至无需以加密形式传送密钥，例如，Diffie-Hellman 技术（有关密钥协议机制的更多信息请参见 ISO/IEC 11770-1）。

3.3.36 object method **对象方法**：可调用资源的一种行为（例如，文件系统可以读、写和执行对象方法）。

3.3.37 one-way function **单向函数**：一个（数学）函数 f，它可以很容易地进行计算，但对范围内的一个普通值 y，很难计算得到 f(x)=y 这种域中的值 x。可能存在少数值 y，对它们来说，计算得到 x 并不困难。

3.3.38 policy mapping **策略映射**：认识到，当一个域中的 CA 对另一个域中的 CA 进行认证时，第一个域中的机构可以认为第二个域中特定的证书策略等同于（不必在所有方面都相同）第一个域中特定的证书策略。

3.3.39 private key; secret key（不赞成的） **专用密钥；秘密密钥**：（在公开密钥加密系统中）用户密钥对中只有用户知晓的那个密钥。

3.3.40 privilege **特权**：由一个机构分配给一个实体的一个属性或特性。

3.3.41 privilege asserter **特权声明者**：一个利用其属性证书或公开密钥证书来声明特权的特权持有者。

3.3.42 privilege management infrastructure（PMI） **特权管理基础设施**：能够支持特权管理、支持综合授权服务、与公开密钥基础设施有关的基础设施。

批注 [P44]: Page: 5 ... 44
批注 [P45]: Page: 5 ... 45
批注 [P46]: Page: 5 ... 46
批注 [P47]: Page: 5 ... 47
批注 [P48]: Page: 5 ... 48
批注 [P49]: Page: 5 ... 49
批注 [P50]: Page: 5 ... 50
批注 [P51]: Page: 5 ... 51
批注 [P52]: Page: 5 ... 52
批注 [P53]: Page: 5 ... 53
批注 [P54]: Page: 5 ... 54
批注 [P55]: Page: 5 ... 55
批注 [P56]: Page: 5 ... 56
批注 [P57]: Page: 5 ... 57
批注 [P58]: Page: 5 ... 58
批注 [P59]: Page: 5 ... 59
批注 [P60]: Page: 5 ... 60
批注 [P61]: Page: 5 ... 61
批注 [P62]: Page: 5 ... 62
批注 [P63]: Page: 5 ... 63
批注 [P64]: Page: 5 ... 64
批注 [P65]: Page: 5 ... 65
批注 [P66]: Page: 5 ... 66
批注 [P67]: Page: 5 ... 67
批注 [P68]: Page: 5 ... 68
批注 [P69]: Page: 5 ... 69
批注 [P70]: Page: 5 ... 70
批注 [P71]: Page: 5 ... 71
批注 [P72]: Page: 5 ... 72
批注 [P73]: Page: 5 ... 73
批注 [P74]: Page: 5 ... 74
批注 [P75]: Page: 5 ... 75
批注 [P76]: Page: 5 ... 76
批注 [P77]: Page: 5 ... 77
批注 [P78]: Page: 5 ... 78
批注 [P79]: Page: 5 ... 79
批注 [P80]: Page: 5 ... 80
批注 [P81]: Page: 5 ... 81
批注 [P82]: Page: 5 ... 82
批注 [P83]: Page: 5 ... 83
... 84
... 85

3.3.43 privilege policy **特权策略：**描述条件的策略，以便特权验证者依据这些条件为合格的特权声明者提供/执行敏感的服务。特权策略与服务相关的属性有关，并与特权声明者相关的属性有关。

3.3.44 privilege verifier **特权验证者：**一个依据特权策略对证书进行验证的实体。

3.3.45 public-key **公开密钥：**（在公开密钥加密系统中）用户密钥对中公开知晓的那个密钥。

3.3.46 public-key certificate（PKC） **公开密钥证书：**用户的公开密钥，以及其他一些信息，利用发放它的认证机构的专用密钥，通过数字签名不可伪造地予以提供。

3.3.47 public key infrastructure（PKI） **公开密钥基础设施：**能够支持公开密钥管理的、能够支持鉴权、加密、完整性或不可否认服务的基础设施。

3.3.48 relying party **信赖方：**在决策中依赖证书中数据的用户或代理。

3.3.49 role assignment certificate **角色分配证书：**包含角色属性的一个证书，它将一个或多个角色分配给证书对象/持有者。

3.3.50 role specification certificate **角色规范证书：**包含分配给某个角色特权指派的一个证书。

3.3.51 sensitivity **灵敏度：**意味着其价值或重要性的一个资源特性。

3.3.52 simple authentication **简单鉴权：**通过简单的口令排列方式进行的鉴权。

3.3.53 security policy **安全策略：**由管理安全服务和工具使用和提供的安全机构设置的规则集。

3.3.54 self-issued AC **自发放的 AC：**一个属性证书，其发放者和对象为同一属性机构。一个属性机构可以使用自发放的 AC，例如，来发布策略信息。

3.3.55 self-issued certificate **自发放的证书：**一个公开密钥证书，其发放者和对象为同一 CA。一个 CA 可以使用自发放的证书，例如，在密钥更新操作期间，来提供从旧密钥到新密钥的信任。

3.3.56 self-signed certificate **自签署的证书：**自发放的证书的一种特殊情况，其中 CA 用于签署对应公开密钥的证书的专用密钥在证书内进行认证。一个 CA 可以使用自签署的证书，例如，来公告其公开密钥或有关其操作的其他信息。

注 — 关于非 CA 发放的自发放证书和自签署证书的使用，不在本建议书 | 国际标准的范畴内。

3.3.57 source of authority（SOA） **机构源：**一个属性机构，某个特殊资源的特权验证者将之看作是分配一系列特权的最终机构。

3.3.58 strong authentication **强鉴权：**通过加密获得的证书方式进行的鉴权。

3.3.59 trust **信任：**一般地，当一个实体（第一个实体）假定第二个实体将按第一个实体所期望的那样准确行事时，那么认为该实体“信任”第二个实体。这种信任只可以应用于某些特定的功能。在本框架内信任的关键作用是描述鉴权实体与机构之间的关系；一个实体必须确认它信任机构能够只创建有效和可靠的证书。

3.3.60 trust anchor **信任锚点：**一个信任锚点是除公开密钥之外的一系列以下信息：算法标识符、公开密钥参数（如果合适的话）、相关专用密钥持有者（即对象 CA）的不同的名称，以及可选的有效期限。信任锚点可以以自签署的证书的形式进行提供。信任锚点为使用证书的系统所信任，并用于验证认证通路中的证书。

4 缩写词

就本建议书 | 国际标准而言，下列缩写词适用：

AA	属性机构
AARL	属性机构撤消清单
AC	属性证书
ACRL	属性证书撤消清单
CA	认证机构
CARL	认证机构撤消清单

批注 [P86]: Page: 6 ... 86
批注 [P87]: Page: 6 ... 87
批注 [P88]: Page: 6 ... 88
批注 [P89]: Page: 6 ... 89
批注 [P90]: Page: 6 ... 90
批注 [P91]: Page: 6 ... 91
批注 [P92]: Page: 6 ... 92
批注 [P93]: Page: 6 ... 93
批注 [P94]: Page: 6 ... 94
批注 [P95]: Page: 6 ... 95
批注 [P96]: Page: 6 ... 96
批注 [P97]: Page: 6 ... 97
批注 [P98]: Page: 6 ... 98
批注 [P99]: Page: 6 ... 99
批注 [P100]: Page: 6 ... 100
批注 [P101]: Page: 6 ... 101
批注 [P102]: Page: 6 ... 102
批注 [P103]: Page: 6 ... 103
批注 [P104]: Page: 6 ... 104
批注 [P105]: Page: 6 ... 105
批注 [P106]: Page: 6 ... 106
批注 [P107]: Page: 6 ... 107
批注 [P108]: Page: 6 ... 108
批注 [P109]: Page: 6 ... 109
批注 [P110]: Page: 6 ... 110
批注 [P111]: Page: 6 ... 111
批注 [P112]: Page: 6 ... 112
批注 [P113]: Page: 6 ... 113
批注 [P114]: Page: 6 ... 114
批注 [P115]: Page: 6 ... 115
批注 [P116]: Page: 6 ... 116
批注 [P117]: Page: 6 ... 117
批注 [P118]: Page: 6 ... 118
批注 [P119]: Page: 6 ... 119
批注 [P120]: Page: 6 ... 120
批注 [P121]: Page: 6 ... 121
批注 [P122]: Page: 6 ... 122
批注 [P123]: Page: 6 ... 123
批注 [P124]: Page: 6 ... 124
批注 [P125]: Page: 6 ... 125
... 126
... 127

CRL	证书撤消清单
dCRL	Delta 证书撤消清单
DIB	号码簿信息库
DIT	号码簿信息树
DSA	号码簿系统代理
DUA	号码簿用户代理
EARL	终端实体属性证书撤消清单
EPRL	终端实体公开密钥证书撤消清单
iCRL	间接的证书撤消清单
OCSP	在线证书状态协议
PKC	公开密钥证书
PKCS	公开密钥密码系统
PKI	公开密钥基础设施
PMI	特权管理基础设施
SOA	机构源

5 惯例

除少数例外，本号码簿规范是根据“*ITU-T | ISO/IEC 通用文本的表述准则*（2001 年 11 月）”的要求制定的。

术语“号码簿规范（或本号码簿规范）”指的是 ITU-T X.509 建议书|ISO/IEC 9594-8。术语“系列号码簿规范”指的是 X.500 系列建议书和 ISO/IEC 9594 的所有部分。

本号码簿规范使用术语“*第 1 版系统*”来指遵循系列号码簿规范第 1 版的所有系统，即 1988 年版本的 CCITT X.500 系列建议书和 ISO/IEC 9594：1990 年版本。本号码簿规范使用术语“*第 2 版系统*”来指遵循系列号码簿规范第 2 版的所有系统，即 1993 年版本的 ITU-T X.500 系列建议书和 ISO/IEC 9594：1995 年版本。本号码簿规范使用术语“*第 3 版系统*”来指遵循系列号码簿规范第 3 版的所有系统，即 1997 年版本的 ITU-T X.500 系列建议书和 ISO/IEC 9594：1998 年版本。本号码簿规范使用术语“*第 4 版系统*”来指遵循系列号码簿规范第 4 版的所有系统，即 2001 年版本的 ITU-T X.500、X.501、X.511、X.518、X.519、X.520、X.521、X.525、X.530 建议书和 2000 年版本的 ITU-T X.509 建议书以及 ISO/IEC 9594：2001 年版本的第 1 到第 10 部分。

本号码簿规范使用术语“*第 5 版系统*”来指遵循系列号码簿规范第 5 版的所有系统，即 2005 年版本的 ITU-T X.500、X.501、X.509、X.511、X.518、X.519、X.520、X.521、X.525 和 X.530 建议书以及 ISO/IEC 9594：2005 年版本的第 1 到第 10 部分。

本号码簿规范使用粗体字体来表示 ASN.1 符号。若在常规文本中要表示 ASN.1 的类型和值时，为了区别于常规文本，使用了粗体字表示。为了表示过程的语义而引用过程名时，为了区别于常规文本，使用了粗体字表示。访问控制许可使用斜体字表示。

如果清单中的各术语是编了号的（使用“—”或字母），那么各术语将被认为是某个程序中的各步骤。

本号码簿规范使用的记法在下面表 1 中定义。

批注 [P128]: Page: 7
A: CRL Certificate Revocation List

批注 [P129]: Page: 7
A: dCRL Delta Certificate Revocation List

批注 [P130]: Page: 7
A: DIB Directory Information Base

批注 [P131]: Page: 7
A: DIT Directory Information Tree

批注 [P132]: Page: 7
A: DSA Directory System Agent

批注 [P133]: Page: 7
A: DUA Directory User Agent

批注 [P134]: Page: 7
A: EARL End-entity Attribute certificate ... 128

批注 [P135]: Page: 7
A: EPRL End-entity Public-key certificate Revocation List ... 129

批注 [P136]: Page: 7
A: iCRL Indirect Certificate Revocation List ... 130

批注 [P137]: Page: 7
A: OCSP Online Certificate Status Protocol ... 131

批注 [P138]: Page: 7
A: PKC Public-Key Certificate ... 132

批注 [P139]: Page: 7
A: PKCS Public-Key Cryptosystem ... 133

批注 [P140]: Page: 7
A: PKI Public-Key Infrastructure ... 134

批注 [P141]: Page: 7
A: PMI Privilege Management Infrastructure ... 135

批注 [P142]: Page: 7
A: SOA Source of Authority

表 1－记法

记　法	含　义
Xp	用户 X 的公开密钥。
Xs	X 的专用密钥。
Xp[I]	利用公开密钥 X，对某些信息 I 加密。
Xs[I]	利用专用密钥 X，对 I 加密。
X{I}	用户 X 签署 I。它由 I 组成，附带一个经过加密的概述。
CA(X)	用户 X 的认证机构。
$CA^n(X)$	（当 n>1 时）：CA（CA（...n 乘以...（X）））
X_1<<X_2>>	由认证机构 X_1 发放的用户 X_2 的证书。
X_1<<X_2>> X_2<<X_3>>	一个证书链（可以是任意长度），其中每个项是认证机构产生下一个项的证书。它在功能上等同于以下证书 X1<<X_{n+1}>>。例如，拥有 A<<B>>B<<C>>将提供等同于 A<<C>>的功能，即在 Ap 前提下找到 Cp 的能力。
X_1p ° X_1<<X_2>>	打开一个证书（或证书链）的操作将提取一个公开密钥。它是一个中缀操作符，其左边操作数为认证机构的一个公开密钥，其右边操作数为该认证机构发放的一个证书。结果是证书为右边操作数的用户的公开密钥。例如： Ap ° A<<B>>　B<<C>> 表示 A 利用公开密钥进行的操作将从其证书获得 B 的公开密钥 Bp，接着利用 Bp 打开 C 的证书。操作的结果是得到 C 的公开密钥 Cp。
A→B	从 A 到 B 的一条认证通路，形成一个证书链，开始于 CA(A)<<CA^2(A)>>，并结束于 CA(B)<<B>>。
注 — 在表中，符号 X、X_1、X_2 等用于替换用户的名称，符号 I 用于替换任意信息。	

6　框架概述

本规范定义了一个用于获得和信任某个实体公开密钥的框架，以便对将要由该实体解密的信息进行加密，或者为了验证该实体的数字签名。框架包括通过某个认证机构（CA）发放一个公开密钥证书，并由证书用户对该证书进行确认。确认包括：

— 在证书用户和证书对象之间建立证书的信任通路；

— 验证通路上每个证书的数字签名；以及

— 确认通路上的所有证书（即在某个特定的时间它们未过期或未被撤消）。

本规范定义了一个用于获得和信任某个特权属性的框架，以便确定是否授权它们访问某个特殊的资源。框架包括通过某个属性机构（AA）发放一个证书，以及由特权验证者对该证书进行确认。确认包括：

— 确保当依据特权策略进行比较时证书中的特权是足够的；

— 如果需要的话，建立一条证书信任委托通路；

— 验证通路上每个证书的数字签名；

— 确保每个发放者有权委托特权；以及

— 确认证书未过期或未被其发放者撤消。

虽然 PKI 和 PMI 是独立的基础设施，并可以相互独立地建立，但它们是相关的。本规范建议，通过其适当公开密钥证书的指针，在属性证书内确定属性证书的持有者和发放者。对属性证书发放者和持有者的鉴权用于确保要求特权和发放特权的实体为提出要求的实体，这通过利用 PKI 的正常程序对身份进行鉴别来实现。在属性证书框架内不重复该鉴权过程。

6.1 数字签名

数字签名用在 PKI 和 PMI 中，发放证书的机构用此机制来确认证书中的绑定。在 PKI 中，公开密钥证书发放 CA 的数字签名用于确认公开密钥素材与证书对象之间的绑定。在 PMI 中，发放 AA 的数字签名用于确认属性（特权）与证书持有者之间的绑定。本子节用于描述一般性的数字签名。本规范的第 2 部分和第 3 部分详细讨论了在 PKI 和 PMI 中数字签名的使用情况。

本子节无意为一般性数字签名规定一个标准，但规定了在 PKI、PMI 和号码簿中签署令牌的方法。

通过将信息添加至一个经过加密的信息摘要来对它进行签署。通过单路散列函数的方法来产生摘要，并利用签署者（见图 1）的专用密钥完成加密。因此：

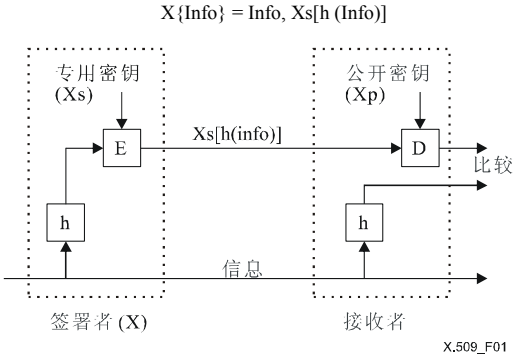


图 1－数字签名

注 1 — 使用专用密钥的加密确保无法伪造签名。散列函数的单路特性确保产生的伪造信息具有相同的散列结果，（因而签名）无法替代。

经签署的信息的接收者通过以下方法对签名进行确认：

— 对信息应用单路散列函数；

— 利用签署者的公开密钥，对结果和通过解密签名得到的结果进行比较。

本规范不要求在签署中使用一个单个的单路散列函数。它有意使框架适用于任何合适的散列函数，并因此支持对方法进行修改，作为未来在密码系统、数学技术或计算能力方面进展的结果。不过，为了正确实施鉴权，希望进行鉴权的两个用户需支持相同的散列函数。因此，在一系列相关应用的范畴内，单个函数的选择应尽可能使更多的用户团体能够实现可靠的鉴权和通信。

经签署的信息包括用于确定散列算法的指示符，以及用于确定用于计算数字签名的加密算法的指示符。

可以利用以下 ASN.1 对某些数据项的加密进行描述：

```
ENCRYPTED { ToBeEnciphered }::=BIT STRING ( CONSTRAINED BY {
    -- 将为把加密程序用于BER编码的八比特组的结果， --
    -- 值为 -- ToBeEnciphered } )
```

通过采用八比特组并将某个加密程序应用于这些八比特组来产生位字符串的值，这些八比特组形成 **ToBeEnciphered** 类型值的完整编码（利用 ASN.1 基本编码规则 — ITU-T X.690 建议书（2002）| ISO/IEC 8825-1：2002）。

注 2 — 加密程序要求就所用的算法达成协定，包括算法的任何参数，如任何必要的密钥、初始化值和填充指令。由加密程序负责规定用于实现数据发送者和接收者同步的方法，它可以包括待发送位中的信息。

注 3 — 要求加密程序输入一个八比特组串，作为输入；并产生一个单个的位串，作为其结果。

注 4 — 数据发送者和接收者关于加密算法及其参数的安全协议机制在本号码簿规范的范畴之外。

某些数据项的签名是通过对缩短的或“散列的”数据项转换结果进行加密而形成的，并可以通过以下 ASN.1 进行描述：

```
HASH {ToBeHashed}       ::=     SEQUENCE {
algorithmIdentifier             AlgorithmIdentifier,
hashValue                       BIT STRING ( CONSTRAINED BY {
        -- 将为把散列程序用于DER编码的八比特组的结果，--
        -- 值为 -- ToBeHashed } ) }

ENCRYPTED-HASH { ToBeSigned }   ::=     BIT STRING ( CONSTRAINED BY {
        -- 将为把散列程序用于DER编码的八比特组的结果，--
        -- 值为 -- ToBeSigned -- 而后把加密程序用于这些八比特组 -- })

SIGNATURE { ToBeSigned }        ::=     SEQUENCE {
        algorithmIdentifier     AlgorithmIdentifier,
        encrypted               ENCRYPTED-HASH { ToBeSigned }}
```

注 5 — 加密程序需要注 2 中所列的协定，并且还需要商定是否直接对散列的八比特组进行加密，或者只有在利用 ASN.1 基本编码规则将它们进一步编码为 **BIT STRING** 后才进行加密。

在签名附加于某个数据类型的情况下，可以使用以下 ASN.1 来定义数据类型，它来自将签名附加于特定的数据类型。

```
SIGNED { ToBeSigned }           ::=     SEQUENCE {
    toBeSigned                  ToBeSigned,
    COMPONENTS OF               SIGNATURE { ToBeSigned }}
```

为了能在分布式环境中实现对 **SIGNED** 和 **SIGNATURE** 类型的验证，需要一种不同的编码。通过应用在 ITU-T X.690 建议书（2002） | ISO/IEC 8825-1：2002 中定义的基本编码规则，将得到 **SIGNED** 或 **SIGNATURE** 数据值的不同编码，它有以下限制：

a） 将使用长度编码的确切形式，以最低数量的八比特组进行编码；

b） 对字符串类型，不得使用编码的构造形式；

c） 如果类型的值为其缺省值，那么它将不出现；

d） 集合类型的部件将按其标签值的升序进行编码；

e） 集合类型的部件将按其八比特组值的升序进行编码；

f） 如果布尔类型的值为 TRUE，那么编码时应将其内容八比特组设为“FF”16；

g） 最终的位字符串值编码八比特组中的每个未用位，如果存在，那么应设为 0；

h） 实数类型的编码应满足以下要求，即基数 8、10 和 16 不得使用，并且二进制比例因子应为 0；

i） UTC 时间的编码应符合 ITU-T X.690 建议书（2002） | ISO/IEC 8825-1：2002 的规定；

j） 通用时间的编码应符合 ITU-T X.690 建议书（2002） | ISO/IEC 8825-1：2002 的规定。

产生一个不同的编码需要数据的抽象语法编码得完全可被理解。可以要求本号码簿对数据进行签署或对包含未知协议扩展或未知属性语法的数据签名进行检查。号码簿将遵循以下规则：

— 它将保留以下接收到信息的编码，即它不完全了解这些信息的抽象句法，并希望之后签署；

— 当签署数据以便发送时，它将发送以下数据，即它完全了解这些信息的句法，信息带一种不同的编码，以及任何其他带其保留的编码的数据，并将对其发送的实际编码进行签署；

— 当对接收数据中的签名进行检查时，它将依据实际接收到的数据对签名进行检查，而不是依据接收数据至不同编码的转换结果。

第 2 部分 — 公开密钥证书框架

此处定义的公开密钥证书框架供有鉴权、完整性、机密性和不可否认要求的应用使用。

机构可以通过一个称为公开密钥证书的数字签署数据结构来将一个公开密钥绑定于一个实体。公开密钥证书的格式在此定义，包括一个扩展性机制和一系列特殊的证书扩展。如果出于某种理由机构撤消一个先前发放的公开密钥证书，那么用户需要能够知道撤消已经发生，从而使之不使用一个不可信赖的证书。撤消清单是一种可用于通知用户有关撤消情况的方案。撤消清单的格式在此定义，包括一个扩展性机制和一系列撤消清单扩展。在证书和撤消清单这两种情况下，其他团体还可以定义额外的、对其特定环境有用的扩展。

使用公开密钥证书的系统在将该证书用于某个应用之前需要对证书进行确认。执行确认任务的程序也在此处定义，包括对证书本身的完整性、其撤消状态、其有关计划用途的有效性等进行验证。

号码簿在提供包括以下内容的安全服务时使用公开密钥证书：

— 在号码簿部件当中进行的强鉴权；

— 号码簿操作的鉴权、完整性和机密性；以及

— 保存数据的完整性和鉴权。

7 公开密钥和公开密钥证书

为使用户能够信任另一个用户的公开密钥，例如，用于鉴别该用户的身份，那么需要从一个信任源获得公开密钥。这样一个称为认证机构（CA）的信任源通过发放一个公开密钥证书来对公开密钥进行认证，公开密钥证书将公开密钥绑定于持有对应专用密钥的实体上。CA 用于确保实体确实拥有专用密钥的程序以及与公开密钥证书发放有关的其他程序超出了本规范的讨论范围。证书具有以下特性，其形式在本节后面规定：

— 有权访问认证机构公开密钥的任何用户都可以恢复经过认证的公开密钥；

— 除了认证机构，任何一方都不可以对没有检测到的证书进行修改（不可伪造的证书）。

由于证书是不可伪造的，因此可以通过将它们置于号码簿中来发布，而无需后者付出特别的努力来保护它们。

注 1 — 虽然通过 DIT 中的一个不同名称对 CA 进行了明确定义，但这并不意味着 CA 的组织机构与 DIT 之间存在任何关系。

认证机构通过签署一系列信息（请参见第 6.1 节）来产生用户的证书，包括用户的不同名称和公开密钥，以及可选的、包含用户额外信息的惟一标识符。惟一标识符内容的准确格式在此未做规定，留待认证机构做出，例如，可能是一个对象标识符、一个证书、一个日期，或者是用于认证不同名称有效性的某种其他形式。尤其是，由认证机构（名称为 CA、惟一标识符为 UCA）产生的用户证书（不同名称为 A、惟一标识符为 UA）具有以下形式：

$$CA\langle\langle A\rangle\rangle = CA\{V,SN,AI,CA,UCA,A,UA,Ap,T^A\}$$

其中：V 为证书版本，SN 为证书序列号，AI 为用于签署证书的算法标识符，UCA 为可选的 CA 惟一标识符，UA 为可选的用户 A 惟一标识符，T^A 指的是证书的有效期，由两个日期组成，在第一个日期和最后一个日期上证书有效。证书有效期指的是以下时间间隔，即在此期间 CA 将保证对证书的状态信息进行维护，即公布撤消数据。由于假定 T^A 将在小于 24 小时的周期内发生变化，因此系统有望使用协调的世界时间作为参考时间基础。任何用户都可以利用 CAp 方面的知识来对证书中签名的有效性进行检查。可用以下 ASN.1 数据类型来表示证书。

```
Certificate                 ::=     SIGNED { SEQUENCE {
   version              [0]     Version DEFAULT v1,
   serialNumber                 CertificateSerialNumber,
   signature                    AlgorithmIdentifier,
   issuer                       Name,
```

```
    validity                    Validity,
    subject                     Name,
    subjectPublicKeyInfo        SubjectPublicKeyInfo,
    issuerUniqueIdentifier  [1] IMPLICIT UniqueIdentifier OPTIONAL,
                                    -- 如果出现，版本应为v2或v3。
    subjectUniqueIdentifier [2] IMPLICIT UniqueIdentifier OPTIONAL,
                                    --如果出现，版本应为v2或v3。
    extensions              [3] Extensions OPTIONAL
                                    --如果出现，版本应为v3。 --} }

    Version                     ::=     INTEGER { v1(0), v2(1), v3(2) }

    CertificateSerialNumber     ::=     INTEGER

AlgorithmIdentifier     ::=     SEQUENCE {
algorithm           ALGORITHM.&id ({SupportedAlgorithms}),
parameters              ALGORITHM.&Type ({SupportedAlgorithms}{ @algorithm}) OPTIONAL }
--   延期定义以下信息对象集，可能是为了
--   标准化概述或规范实施一致性声明。
--   要求信息对象集规定一个表，以约束AlgorithmIdentifier的parameters部件。
--   SupportedAlgorithms        ALGORITHM   ::=     { ... }

Validity                    ::=         SEQUENCE {
    notBefore       Time,
    notAfter        Time }

SubjectPublicKeyInfo                    ::=         SEQUENCE {
    algorithm               AlgorithmIdentifier,
    subjectPublicKey        BIT STRING }

Time  ::=  CHOICE {
    utcTime             UTCTime,
    generalizedTime     GeneralizedTime }

Extensions ::= SEQUENCE OF Extension

Extension ::=  SEQUENCE {
    extnId          EXTENSION.&id ({ExtensionSet}),
    critical        BOOLEAN DEFAULT FALSE,
    extnValue       OCTET STRING
                    -- 包含一个类型值的DER编码以及
                    -- 由extnId 确定的扩展对象的ExtnType。

ExtensionSet    EXTENSION       ::=     { ... }
```

在 **Time** 值用于任何比较操作之前，例如，作为搜索中匹配规则的一部分，并且如果 **Time** 的语法已经选定为 **UTCTime** 类型，那么两个数字位的年份字段的值将阐释为四个数字位的年份值，如下所示：

— 如果 2 个数字位的值为 00-49（包含之），那么值将为 2000 加 2 位数字值。

— 如果 2 个数字位的值为 50-99（包含之），那么值将为 1900 加 2 位数字值。

注 2 — 使用 **GeneralizedTime** 可以防止与不知选择 **UTCTime** 还是 **GeneralizedTime** 可能性的执行方案互联。由规定域的负责者来确定何时可以使用 **GeneralizedTime**，在该域中使用在本号码簿规范中定义的证书，如概况组。任何情况下 **UTCTime** 都不得用于表示 2049 年之外的日期。

Version 为经过编码的证书的版本。如果在证书中出现 **extensions** 部件，那么版本应为 v3。如果出现 **issuerUniqueIdentifier** 或 **subjectUniqueIdentifier** 部件，那么版本应为 v2 或 v3。

serialNumber 是 CA 为每个证书分配的一个整数。对某个特定 CA 发放的每个证书，**serialNumber** 的值应是惟一的（即利用发放者姓名和序列号可以确定一个惟一的证书）。

Signature 包含签署证书过程中所用的算法和散列函数的算法标识符（例如，md5WithRSAEncryption、sha-1WithRSAEncryption、id-dsa-with-sha1 等）。

Issuer 用于确定签署和发放证书的实体。

Validity 是一个时间间隔，在其期间 CA 保证，它将对证书状态信息进行维护。

Subject 用于确定与在对象公开密钥字段中找到的公开密钥相关的实体。

subjectPublicKeyInfo 用于传达正在经受认证的公开密钥，并用于确定该公开密钥为其一个实例的算法（例如，rsaEncryption、dhpublicnumber、id-dsa 等）。

在名称重用情况下，**issuerUniqueIdentifier** 用于惟一确定一个发放者。

在名称重用情况下，**subjectUniqueIdentifier** 用于惟一确定一个对象。

注 3 — 在不同名称可能由命名机构重新指派给一个不同用户的情况下，CA 可以使用惟一标识符在重用的实例之间做出区分。不过，如果由多个 CA 向同一用户提供证书，那么建议作为其用户注册程序的一部分，CA 应对惟一标识符的指派进行协调。

extensions 字段允许向结构增加新的字段，而无需对 ASN.1 定义进行修改。一个扩展字段由一个扩展标识符、一个关键性标志、一个与确定的扩展有关的 ASN.1 类型数据值编码组成。对 **SEQUENCE** 中单个扩展排序至关重要的那些扩展来说，对这些单个扩展的规定应包括有关当中排序重要性的规则。当处理证书的执行方案不认可某个扩展时，如果关键性标志为 **FALSE**，那么它可以忽略该扩展。如果关键性标志为 **TRUE**，那么未被认可的扩展将使得结构被认为是非有效的，即在证书中，未被认可的关键扩展将造成使用该证书的签署验证失败。当使用执行方案的证书认可并能处理扩展时，使用执行方案的证书将对扩展进行处理，而不管关键性标志的值是什么。注意：任何标志为非关键的扩展都将在处理扩展的使用证书系统与不认可扩展并忽略它的使用证书系统之间产生不一致行为。

如果在扩展中出现未知的元素，并且扩展为标志为关键的，那么将依据 ITU-T X.519 建议书 | ISO/IEC 9594-5 第 12.2.2 节中所述的扩展性规则，忽略这些未知的元素。

对一个扩展，CA 可以有三个选项：

i） 它可以从证书中排除扩展；

ii） 它可以包括扩展，并将之标志为非关键的；

iii） 它可以包括扩展，并将之标志为关键的。

对一个扩展，确认机制可以采取两种可能的措施：

i） 它可以忽略扩展，并接受证书（所有其他事情都一样）；

ii） 它可以处理扩展，并依据扩展的内容以及处理的执行条件（例如，通路处理变量的当前值），接受或拒绝证书。

对某些扩展只能标志为关键的。在这些情况下，一个对扩展理解的验证机制将对它进行处理，接受/拒绝证书取决于（至少部分地取决于）扩展的内容。不理解扩展的验证机制将拒绝证书。

对某些扩展只能标志为非关键的。在这些情况下，一个对扩展理解的验证机制将对它进行处理，接受/拒绝证书取决于（至少部分地取决于）扩展的内容。不理解扩展的验证机制将接受证书（除非该扩展以外的其他因素导致它被拒绝）。

对某些扩展可以标志为关键的，也可以标志为非关键的。在这些情况下，一个对扩展理解的验证机制将对它进行处理，接受/拒绝证书取决于（至少部分地取决于）扩展的内容，而不管关键性标志是什么。如果扩展被标志为非关键的，那么不理解扩展的验证机制将接受证书（除非该扩展以外的其他因素导致它被拒绝），如果扩展被标志为关键的，那么拒绝证书。

当 CA 考虑在证书中纳入一个扩展时，它希望在任何可能的地方都坚持其意图。如果在对证书表示信任之前需要对扩展的内容进行考虑，那么 CA 将把扩展标志为关键的。这么做需要认识到任何不对扩展做处理的验证机制将拒绝证书（可能限制能对证书进行验证的应用集）。CA 可以将某个扩展标志为非关键的，以便实现与不能处理扩展之确认应用的向后兼容。当有关向后兼容性以及与不能处理扩展之确认应用间互操作性的需求比 CA 增强扩展之能力更重要时，这些可选的关键扩展将被标志为非关键的。在转换周期期间，当验证者的认证处理应用升级为能够对扩展进行处理时，CA 很可能会把可选的关键扩展设置为非关键的。

可以在 ITU-T 建议书 | 国际标准中来确定特殊的扩展，或通过有需求的任何组织机构来确定。将依据 ITU-T X.660 建议书 | ISO/IEC 9834-1，来定义用于确定扩展的对象标识符。证书的标准扩展在本号码簿规范的第 8 节中进行定义。

以下对象类别用于定义特定的扩展。

```
EXTENSION ::= CLASS {
    &id     OBJECT IDENTIFIER UNIQUE,
    &ExtnType }
WITH SYNTAX {
    SYNTAX                  &ExtnType
    IDENTIFIED BY           &id }
```

有两种主要类型的公开密钥证书，即终端实体证书和 CA 证书。

一个终端实体证书是一个由 CA 发放给某个对象的证书，该对象不是其他公开密钥证书的发放者。

一个 CA 证书是一个由 CA 发放给某个对象的证书，该对象本身也是一个 CA，因此它能够发放公开密钥证书。CA 证书本身可以通过以下类型进行分类：

— 自发放证书 — 这是这样一个证书，即它的发放者和对象为同一 CA。CA 可以使用自发放证书，例如，在密钥更新操作期间，提供从旧密钥到新密钥的信任。

— 自签署证书 — 这是自发放证书的一个特例，当中 CA 用于签署证书的专用密钥对应在证书中进行认证的公开密钥。CA 可以使用自发放证书，例如，用于公告其公开密钥或其他有关其操作的信息。

— 交叉证书 — 这是这样一个证书，即它的发放者和对象为不同的 CA。CA 向其他 CA 发放证书，可以作为一种授权对象 CA 存在的机制（例如，在严格的层次结构中），或者用于认可对象 CA 的存在（例如，在分布式信任模型中）。对这两种情况都可以使用交叉证书结构。在某些情况下，有关约束的冲突或重叠要求，如名称约束，可能需要一个 CA 向另一个 CA 发放多个交叉证书。

各用户 A（参与强鉴权）的号码簿条目包含 A 的证书。由 A 的认证机构来产生这样一个证书，它是 DIT 中的一个条目。A 的认证机构（可能不是惟一的）表示为 CA(A)，如果了解 A，那么可以简单地表示为 CA。那么知道 CA 公开密钥的任何用户都可以找到 A 的公开密钥。找到公开密钥因此是递归的。

如果试图获得用户 B 公开密钥的用户 A 已经获得 CA(B)的公开密钥，那么处理完成。为了使 A 能够获得 CA(B)的公开密钥，各认证机构 X 的号码簿条目都包含很多证书。这些证书有两种类型。第一种是其他认证机构产生的 X 前向证书；第二种是 X 本身产生的逆向证书，它们是其他认证机构的、经过认证的公开密钥。这些证书的存在使用户能够构建从一个点到另一个点的认证通路。

允许一个特殊用户获得另一个用户公开密钥的证书清单即*认证通路*。清单中的每一项都是清单中下一项的认证机构的一个证书。从 A 到 B 的一个认证通路（表示为 A→B）将：

— 以 CA(A)产生的一个证书开始，即对某个实体 X1，CA(A)<<X1>>；

— 以更多的证书 Xi<<Xi+1>>继续；

— 以证书 B 结束。

各证书的 **issuer** 和 **subject** 字段部分地用于确定一条有效的通路。对有效认证通路中的每对邻近证书，证书中 **subject** 字段的值应匹配于后续证书中 **issuer** 字段的值。另外，第一个证书中 **issuer** 字段的值应匹配于信任锚点的 DN。当对认证通路的有效性进行检查时，仅使用这些字段中的名称。证书扩展中的名称不用于此目的。逻辑上，一条认证通路在希望鉴权的两个用户之间形成一条不可断的号码簿信息树信任点链。用户 A 和用户 B 用于获得认证通路 A→B 和 B→A 的准确方法可以不同。推动此项工作的一个方法是设计一个有关 CA 的层次结构，它与 DIT 层次结构的全部或部分可以也可以不一致。这样做的好处是，在层次结构中拥有 CA 的各用户可以利用号码簿并无需任何预先信息，来在其间建立一条认证通路。为了实现之，每个 CA 都可以保存一个证书，以及一个用于对应其上级 CA 的逆向证书。应使用在 ITU-T X.501 建议书 | ISO/IEC 9594-2 中定义的 **distinguishedNameMatch** 匹配规则，来对一个证书 **issuer** 字段中的不同名称（DN）与另一个证书 **subject** 字段中的 DN 进行比较。

一个用户可以从一个或多个认证机构获得一个或多个证书。每个证书都带发放它的认证机构的名称。以下 ASN.1 数据类型可用于表示证书和认证通路。

```
Certificates                    ::=     SEQUENCE {
    userCertificate                     Certificate,
    certificationPath                   CertPath OPTIONAL }

CertificationPath               ::=     SEQUENCE {
    userCertificate                     Certificate,
    theCACertificates                   SEQUENCE OF CertificatePair OPTIONAL }
```

另外，以下 ASN.1 数据类型可用于代表前向认证通路。本部件包含可回指向发起者的认证通路。

```
CertPath    ::=     SEQUENCE OF CrossCertificates

CrossCertificates           ::=     SET OF Certificate

PkiPath     ::=     SEQUENCE OF Certificate
```

PkiPath 用于表示一个认证通路。在序列内，证书的次序是这样的，即第一个证书的对象为第二个证书的发放者，等等。

认证通路中的每个证书都将是惟一的。在 **CertificationPath** 部件 **theCACertificates** 的值中，或者在 **CertPath** 部件 **CrossCertificates** 的 **Certificate** 值中，或者在 **PkiPath** 的 **Certificate** 值中，任何证书都不得出现多次。

7.1 密钥对的产生

执行方案总的安全管理策略将对密钥对的生命周期进行定义，但这超出了本框架的范围。不过，保证只有其所属的用户知道所有的专用密钥对总的安全是至关重要的。

对人而言，密钥数据是不容易记住的，因此，应采用一种合适的、便于传送的方法来保存它。一种可能的机制是使用“智能卡”。它将持有用户的专用密钥和（可选的）公开密钥、用户证书，以及认证机构公开密钥的拷贝。此外，例如，应至少使用一个个人身份号码（PIN）来进一步保证智能卡使用的安全，通过要求用户持有智能卡并知道如何访问它来提高系统的安全性。不过，选择用来保存此类数据的准确方法已超出了本号码簿规范的讨论范围。

可以有三种产生用户密钥对的方法：

a） 密钥对由用户自己产生。这种方法的好处是用户的专用密钥绝不会泄露给另一个实体，但要求用户具备一定等级的资格。

b） 密钥对由第三方产生。第三方将用一种在物理上安全的方式来把专用密钥告知用户，而后主动销毁所有与密钥对创建以及密钥对本身有关的信息。应采取合适的、物理上安全的措施来确保第三方和数据操作免遭破坏。

c） 密钥对由 CA 产生。这是 b）的一个特例，应做仔细考虑。

注 — 认证机构已经展示了与用户有关的信任功能，将受制于必要的物理安全措施。该方法具有不需要向 CA 传送安全数据以供认证的优点。

所用的密钥系统对密钥产生会施加特殊的（技术）约束。

7.2 公开密钥证书的创建

一个公开密钥证书与公开密钥以及其所描述的用户的惟一的、不同的名称有关。因此：

a） 在为用户创建证书之前，认证机构应满足用户的身份要求；

b） 认证机构不用以相同名称为两个用户发放证书。

保证向认证机构的信息传送不受到破坏这点很重要，为此应采取适当的物理安全措施。在这方面：

a） 如果 CA 向一个公开密钥已遭破坏的用户发放证书，那么这将是对安全性的严重破坏。

b） 如果采用了第 7.1 节 b）或第 7.1 节 c）中的密钥对产生方法，那么应以一种安全的方式来将用户的专用密钥传送给用户。

c） 如果采用了第 7.1 节 a）或第 7.1 节 b）中的密钥对产生方法，那么用户可以使用不同的方法（在线的或离线的）来以一种安全的方式将其公开密钥传送给 CA。在线方法可以为在用户与 CA 之间执行的远程操作提供一些额外的灵活性。

一个公开密钥证书是一段可公用的信息，对其传送给号码簿无需采取特殊的安全措施。由于它由离线的认证机构代表用户（将向用户提供一个它的拷贝）产生，因此用户只需在一次后续的、对号码簿的访问中，将该信息保存在其号码簿条目中就行了。可选地，CA 可以为用户寄存证书，在这种情况下，应为该代理提供合适的访问权限。

7.3 证书有效性

发放证书（公开密钥或属性）的机构还有责任指明其发放之证书的有效性。通常，证书有可能在之后被撤消。这种撤消以及撤消通知，可以直接地由发放证书的同一机构来完成，或者间接地由发放证书的机构适当授权了的另一个机构来完成。要求发放证书的机构声明其行动，可能通过一份公布的声明，或者通过证书本身，或者通过某种其他确定的方式，是否：

— 不能撤消证书；或者

— 可以通过同一证书发放机构直接撤消证书；或者

— 证书发放机构授权一个不同的实体来执行撤消。

要求执行证书撤消任务的机构通过某种类似的方式声明，信赖方可以使用什么机制来获得有关该机构发放之证书的撤消状态信息。本规定定义了一种证书撤消清单（CRL）机制，但不排除使用其他可选的机制。一种这样的可选机制是在 IETF RFC 2560[1)] 中规定的在线证书状态协议（OCSP）。使用该协议，一个信赖方（客户机）从一个 OCSP 服务器处请求证书的撤消状态。服务器可以使用 **CRL** 或其他机制来检查证书的状态，并对客户机做出相应的响应。如果信赖方能够使用 OCSP 来检查证书的状态，那么 IETF RFC 3280[2)] 包含一个证书扩展（机构信息访问），它将包括在此类证书中，并将提供足够的信息来访问一个适当的 OCSP 服务器。合适的话，信赖方将对所有在第 10 节所述的通路处理程序和第 16 条所述的委托通路处理程序期间需要考虑的证书检查撤消状态信息，以便对证书进行验证。

只有一个有权发放 CRL 的 CA 可以选择将该机构委托给另一个实体。如果进行了这种委托，那么它在验证证书/CRL 之时应是可验证的。**cRLDistributionPoints** 扩展可供此目的使用。该扩展的 **cRLIssuer** 字段将位于任何实体的名称中，而不是证书发放者本身，这些实体有权发放涉及所议证书撤消状态的 CRL。

证书，包括公开密钥证书以及属性证书，将在其到期之时拥有一个与其相关的生命周期。为了保证服务的连续性，机构应及时保证替换证书的可用性，以替代已经到期/将要到期的证书。撤消通知日期为证书撤消通知首次出现在 CRL 中的日期/时间，而不管它是一个基础 CRL 还是一个 dCRL。在 CRL 中，撤消通知日期是 **thisUpdate** 字段中所包含的值。撤消日期是 CA 实际撤消证书的日期/时间，它可以不同于它出现在 CRL 中的第一个时间。在 CRL 中，撤消日期是 **revocationDate** 部件中所包含的值。无效日期是知道或怀疑专用密钥受到破坏的日期/时间，或者证书被认为无效的日期/时间。该日期可能早于撤消日期。在 **CRL** 中，无效日期是 **invalidityDate** 条目扩展中所包含的值。

两个相关点为：

— 可以对证书的有效性进行设计，以便每个证书能在其先行者到期之时变得有效，或允许重叠。后者将使机构无需安装和分发大量可能在同一终止日期变得无效的证书。

— 通常会从号码簿中移去到期的证书。这是一个有关安全策略的问题，如果规定不可否认数据服务，那么机构有责任将旧的证书再保留一段时间。

1） IETF RFC 2560，*X.509*国际互联网公开密钥基础设施在线证书状态协议（*OCSP*），1999年6月。

2） IETF RFC 3280，国际互联网*X.509*公开密钥基础设施证书和证书撤消清单（*CRL*）概况，2002年4月。

可以在其终止日期之前撤消证书，例如，如果认为用户的专用密钥遭到了破坏，或者机构不再认可用户，或者如果认为机构证书遭到了破坏。机构将告知撤消用户证书或机构证书，合适的话，将使一个新的证书变得可用。而后机构可能通过某种离线程序，将有关撤消信息告知证书持有者。

机构发放并在随后撤消证书：

a） 可以要求它为所有由该机构发放的证书类型保留一份有关其撤消事件的审计记录（例如，发放给终端实体以及其他机构的公开密钥证书、属性证书）；

b） 将向使用 CRL、在线证书状态协议或用于公布撤消状态信息的某种其他机制的各信赖方提供撤消状态信息；

c） 如果使用 CRL，将保留和公布 CRL，即使撤消证书清单为空；

d） 如果只使用经过分割的 CRL，那么将发放一个有关分割 CRL 的全集，涵盖完整的证书集，将使用 CRL 机制来报告其撤消状态。因此，如果 CRL 发放者不使用经过分割的 CRL，那么分割 CRL 的完整集将等同于一个有关同一证书集的完全 CRL。

信赖方可以使用众多机制来确定机构提供的撤消状态信息。例如，在证书本身中可以存在一个指针，将信赖方指向某个位置，在此位置提供撤消信息。在撤消清单中可以存在一个指针，重新将信赖方指向一个不同的位置。信赖方可以在一个知识库（例如，一个号码簿）中或通过在本规范范围之外的其他方式（例如，本地配置）来确定撤消信息。

对受机构撤消清单影响的各号码簿条目进行维护是号码簿及其用户的责任，它们将依据安全策略采取行动。例如，用户可以通过用新的证书替换旧的证书来对其对象条目进行修改。而后将使用新的证书来对号码簿用户进行鉴别。

如果在号码簿中公布撤消清单，那么在条目中持有它们，作为以下类型的属性：

— 证书撤消清单；

— 机构撤消清单；

— Delta 撤消清单；

— 属性证书撤消清单；

— 属性机构撤消清单。

```
CertificateList             ::=     SIGNED { SEQUENCE {
    version                         Version OPTIONAL,
                                    -- 如果出现，版本须为 v2。
    signature                       AlgorithmIdentifier,
    issuer                          Name,
    thisUpdate                      Time,
    nextUpdate                      Time OPTIONAL,
    revokedCertificates             SEQUENCE OF SEQUENCE {
        serialNumber                        CertificateSerialNumber,
        revocationDate                      Time,
        crlEntryExtensions              Extensions OPTIONAL } OPTIONAL,
    crlExtensions           [0]     Extensions OPTIONAL }}
```

Version 为经过编码的撤消清单的版本。如果在撤消清单中出现标志为关键的 **extensions** 部件，那么版本应为 v2。如果在撤消清单中没有出现任何标志为关键的 **extensions** 部件，那么版本要么不存在，或者出现为 v2。

Signature 包含机构用于签署撤消清单的算法的算法标识符。

issuer 用于确定签署和发放撤消清单的实体。

thisUpdate 为发放本撤消清单的日期/时间。

nextUpdate 如果出现，那么指明将发放本系列下一个撤消清单的日期/时间。下一个撤消清单可以在指明的日期之前发放，但不得在任何指明的时间之后发放。

revokedCertificates 用于确定已经撤消的证书。撤消的证书有其序列号进行确定。如果本 CRL 所涵盖的证书没有一个被撤消，那么强烈建议从 CRL 中删去 **revokedCertificates** 参数，而不包括在一个空的 **SEQUENCE** 中。

crlExtensions 如果出现，那么包含一个或多个 CRL 扩展。

注 1 — 检查整个证书清单是一个本地问题。不得假定清单以某种特殊次序排序，除非发放机构已经规定了特殊的排序规则，例如，在该机构的策略中。

注 2 — 如果数据服务的不可否认性依赖于机构提供的密钥，那么服务应确保由一个当前的机构来对机构所有的相关密钥（已撤消的或已到期的）和已做了时间戳记的撤消清单进行归档和认证。

注 3 — 如果包括在 **CertificateList** 中的任何扩展均被定义为关键的，那么 **CertificateList** 的版本元素应出现。如果未包括任何定义为关键的扩展，那么版本元素可以不出现。如果 **version** 未出现，那么可能允许以下执行方案，即如果在对 CRL 中的 **revokedCertificates** 序列进行检查时未遇到扩展，那么只支持第 1 版本 CRL 继续使用 CRL。在未出现版本的情况下，如果能在处理过程早期就确定在 CRL 中未出现任何关键的扩展，那么支持第 2 版本（或更高版本）CRL 的执行方案还可以对其处理过程进行最优化。

注 4 — 当处理证书撤消清单的执行方案不认可 **crlEntryExtensions** 字段中的关键扩展时，它将假设至少确定的证书已被撤消，不再有效，并执行本地策略规定的、有关撤消证书的额外行动。当执行方案不认可 **crlExtensions** 字段中的关键扩展时，它将假设确定的证书已被撤消，不再有效。不过，在后一种情况下，由于清单可能不完整，尚未确定撤消的证书不能假设是有效的。在这种情况下，本地策略将规定需要采取的行动。任何情况下，本地策略都可以规定本规范所述之行动之外的行动，或者比本规范所述之行动更强劲的行动。

注 5 — 如果扩展对清单的处理有影响（例如，需要对多个 CRL 进行扫描，以便实现对整个撤消证书清单的检查，或者一个条目可以代表多个证书），那么在 **crlExtensions** 字段中该扩展将被指为关键的，而不管该扩展在 CRL 中置于何处。在条目 **crlEntryExtensions** 字段中指明的扩展将置于该条目中，并将只对该条目中所规定的各证书有影响。

注 6 — CRL 的标准扩展在本号码簿规范的第 8 节中定义。

如果在扩展中出现了未知的元素，并且扩展未被标志为关键的，那么依据 ITU-T X.519 建议书 | ISO/IEC 9594-5 第 12.2.2 节中所述的扩展性规则，这些未知的元素将被忽略。

7.4 拒绝数字签名

某个事件的任何参与者都可以在之后决定否认参与者在该事件中数字签署的任何事情。例如，一个参与者可以反驳参与了密钥建立或者是某个已签署电子邮件消息的发起者，同样，一个参与者可以反驳签署了某个文档，目的旨在绑定于该文档的内容。否认是不可能成功的。不可否认框架 ITU-T X.813 建议书 | ISO/IEC 10181-4 描述了一个争议解决过程，如下所述：

1） 产生证据；
2） 传送、保存和检索证据；
3） 确认证据；以及
4） 解决争议。

产生的证据可以包括但不限于：

— 与事件相关的审计记录和目的声明；
— 第三方公证人所做的声明；
— 政策声明；
— 数字签署的信息，包括审计记录和公证人声明；
— 数字签署信息的时戳；
— 支持数字签名的证书；
— 公布的且在争议事件发生之时可用的、适当的撤消信息；以及
— 事件发生之后的任何证书撤消，指明事件发生之前存在的关键协议。

对可以作为证据提交的保存数据的完整性，可以通过多种方式进行维护，例如，访问控制、由信任的第三方保存散列、数字签名。还有必要定期地增强对该保存数据的保护，以应对在计算机处理与/或密码分析方面的技术进步。

注 — 产生证据的类型和数量以及完整性的水平都不由本号码簿规范规定。不过，希望努力程度将与所涉风险相称。

证据确认可能需要重新验证数据数字签名的有效性，例如，消息、文档、证书、CRL，以及在最初的验证处理过程中使用的时戳。事实上，已经到期的证书不得阻止将之用于重新验证在该证书有效期期间创建的签名。如果可以确定在争议事件发生之时证书是有效的，那么可以使用已经撤消的证书。

即使上面所述的所有数字证据在技术上认为都是有效的，其他条件，例如，企图、理解或签署者的能力，也可以允许签署者成功否认它。

8 公开密钥证书和CRL扩展

在本节中定义的证书扩展供公开密钥证书使用，除非另有规定。供属性证书使用的扩展在第 15 节中定义。在本节中定义的 CRL 扩展可以用在 CRL、CARLs 中，还可以用于在第 17 节中定义的 ACRL 和 AARLs。

本节用于规定以下领域中的扩展：

a） *密钥和策略信息*：这些证书和 CRL 扩展用于传达有关所涉密钥的额外信息，包括对象和发放者密钥的密钥标识符、计划或限制密钥用法的指示符、证书策略的指示符。

b） *对象和发放者属性*：这些证书和 CRL 扩展为证书对象、证书发放者或 CRL 发放者提供各种不同名称形式的可选名称的支持。这些扩展还可以用于传达有关证书对象的额外信息，有助于证书用户相信证书对象是一个特殊的人或实体。

c） *认证通路约束*：这些证书扩展允许在 CA 证书（即其他 CA 为 CA 发放的证书）包括约束规定，以便在涉及多个证书策略时有助于证书通路的自动处理。当对环境中的不同应用有不同的策略时，或者当发生与外部环境的互操作时，将出现多个证书策略。约束可以限制对象 CA 可以发放的证书类型，或者在后续的认证通路中可以出现的证书类型。

d） *基本的 CRL 扩展*：这些 CRL 扩展允许 CRL 包括撤消理由指示，以便规定临时终止某个证书，以及包括 CRL 发放序列号，以便允许证书用户在某个 CRL 发放者的序列中检测丢失的 CRL。

e） *CRL 分发点和 delta-CRL*：这些证书和 CRL 扩展允许将来自某个 CA 的完整撤消信息集划分至单独的 CRL 中，并允许将来自多个 CA 的撤消信息结合进一个 CRL 中。这些扩展还支持使用部分 CRL，它们只指明自前一个 CRL 发放以来的改变。

是否在证书或 CRL 中包括某个扩展由发放证书或 CRL 的机构来决定。

在证书或 CRL 中，扩展可以标志为关键的或非关键的。如果扩展标志为关键的，并且使用证书的系统不认可扩展字段类型，或者不执行扩展的语义，那么该系统将认为证书是无效的。如果扩展标志为非关键的，那么不认可或不执行该扩展类型的使用证书的系统可以对忽略扩展的证书剩余部分进行处理。如果扩展标志为非关键的，认可该扩展类型的使用证书的系统将对扩展进行处理。本号码簿规范中的扩展类型定义指明，扩展是否总为关键、总为非关键，或者是否可以由证书或 CRL 发放者来决定扩展的关键性。要求某些扩展总为非关键的理由是为了允许使用证书的执行方案（它们无需使用此类扩展）可以不支持它们，而无损与所有认证机构之间的互操作能力。

注 — 使用证书的系统可能需要在证书中出现某些非关键的扩展，以便认为该证书是可接受的。需要包括此类扩展可以通过证书用户的本地策略规则来间接表达，或者可以是一个 CA 策略规则，向使用证书的系统指明，需要在证书策略扩展中包括一个特殊的证书策略标识符，并将扩展标志为关键的。

对在本号码簿规范中定义的所有证书扩展、CRL 扩展和 CRL 条目扩展，对在任何证书、CRL 或 CRL 中的每种扩展类型，均不得存在多个实例。

8.1 策略处理

8.1.1 证书策略

本框架包含三种类型的实体：证书用户、认证机构和证书对象（或终端实体）。各实体依据相对其他两个实体的职责开展工作，反过来，也享用它们提供的有效保证。这些职责和保证在证书策略中定义。证书策略是一个文件（通常以明文形式存在）。它可以通过惟一标识符来引用，可以包括在认证机构发放之证书的证书策略扩展中，证书发放给终端实体，证书用户也有赖于这些证书。可以依据一个或多个策略来发放证书。策略定义和标识符指派由策略机构来完成。策略机构监管的策略集称为策略域。即使策略未在任何地方进行记录，或者未在证书

中引用，所有的策略也都需要依据策略进行发放，本规范不对证书策略的格式或内容做规定。

通过以下行为，证书用户可以完成其在证书策略下的职责，即输入一个机构公开密钥，并将之用作信任锚点，或者依靠一个包括相关策略标识符的证书。通过以下行为，认证机构可以完成其在证书策略下的职责，即发放一个包括相关策略标识符的证书。通过以下行为，终端实体可以完成其在证书策略下的职责，即请求并接受一个包括相关策略标识符的证书，并使用对应的专用密钥。不使用证书策略扩展的执行方案应通过某种其他方式来完成要求的职责。

对简单宣布遵守策略的实体而言，通常不满足框架中其他实体的保证要求。它们需要某种理由来相信其他各方执行的是一个可靠的策略方案。不过，如果在策略中明确做出这种声明，那么策略用户可以接受认证机构的保证，即其终端实体同意担负其在策略下的职责，而无需它们直接进行确认。有关证书策略的这方面问题已超出了本规范的讨论范围。

认证机构可以对其证书的使用施加限制，以便对它认为因发放证书而可能带来的风险进行控制。例如，它可以限制证书用户的团体、这些团体使用其证书的目的，与/或在它自身或其终端实体出现故障的情况下，它准备应对的损坏类型和内容。这些问题应在证书策略中予以定义。

有助于受影响的实体更好地理解策略规定的额外信息，可以以策略限定符的形式，包括在证书策略扩展中。

8.1.2 交叉认证

一个认证机构可以是另一个认证机构所发放证书的对象。在这种情况下，证书称为交叉证书，为证书对象的认证机构称为对象认证机构，发放交叉证书的认证机构称为中间认证机构（见图 2）。交叉证书和终端实体证书都可以包含一个证书策略扩展。

对象认证机构、中间认证机构和证书用户共享的保证和职责通过在交叉证书中确定的证书策略来定义，依据之，对象认证机构可以作为或代表一个终端实体。证书对象、对象认证机构和中间认证机构共享的保证和职责通过在终端实体证书中确定的证书策略来定义，依据之，中间认证机构可以作为或代表一个证书用户。

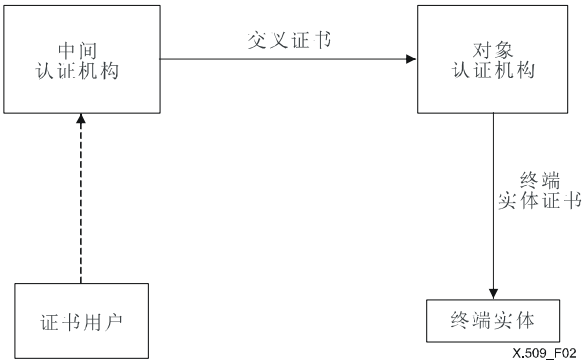


图 2－交叉认证

对公用于通路中所有证书的策略集，认为认证通路是有效的。

一个中间认证机构可以依次成为另一个认证机构发放的证书的对象，因此需要创建长度大于两个证书的认证通路。并且由于随着证书通路长度的增长，信任将受到冲击，需要实施控制来确保证书用户将拒绝相关信任水平

低得无法接受的终端实体证书。这是认证通路处理程序的部分功能。

除了以上所述的情况，还有两种特殊的情况需要考虑：

a） 认证机构不用证书策略扩展来向其证书用户传达它的策略要求；以及

b） 证书用户或中间认证机构将控制策略工作委托给通路中的下一个机构。

在第一种情况下，证书根本不应包含证书策略扩展。结果是，其下通路有效的策略集将为空。但尽管如此，通路仍可为有效的。证书用户将继续确保能够依据通路中机构的策略使用证书。

在第二种情况下，证书用户或中间认证机构应在 *initial-policy-set* 或交叉证书中包括特殊值 *any-policy*。当证书包括特殊值 *any-policy* 时，它不应包括任何其他的证书策略标识符。标识符 *any-policy* 不应拥有任何相关的策略限定符。

证书用户可以确保依据标准（通过设置 *initial-explicit-policy* 指示符）来传达其所有的职责。以这种方式，只有那些将标准认证策略扩展作为其获得绑定方式的机构，才能在通路中被接受，证书用户没有任何额外的职责。由于机构在作为证书用户或代表证书用户时也承担职责，因此它们可以确保依据标准（通过在交叉证书中设置 **requireExplicitPolicy**）来传达其所有的职责。

8.1.3 策略映射

某些认证通路可能跨越各策略域之间的边界。保证和职责（依据它们发放交叉证书）实际上可以等同于某些或全部以下保证和职责，即依据这些保证和职责，对象认证机构向终端实体发放证书，即使对这些实际上等同的策略，策略机构（两个认证机构工作于其下）可能拥有选定的、不同的惟一标识符。在这种情况下，中间认证机构可以在交叉证书中包括一个策略映射扩展。在策略映射扩展中，中间认证机构向证书用户保证，它将继续遵循熟知的保证，并将继续完成其熟知的职责，即使认证通路中的各后续实体工作于一个不同的策略域中。对策略（依据它们发放交叉证书）子集中的每一个，中间认证机构都应包括一个或多个映射。如果一个或多个证书策略（对象认证机构依据之开展工作）等同于以下策略，即中间认证机构依据它们开展工作（即拥有相同的惟一标识符），那么这些标识符不应包括在策略映射扩展中，但可包括在证书策略扩展中。

策略映射的作用是将认证通路更底层证书中的所有策略标识符转换为等同策略的标识符，证书用户认可这么做。

策略不得映射至特殊的值 *any-policy* 或者自特殊的值 *any-policy* 映射。

证书用户可以决定不应依赖在策略域中发放的而不是它自己发放的证书，即使一个信任的中间认证机构可以决定其策略实际上等同于其自身。可以通过将 *initial-policy-mapping-inhibit input* 设为通路验证程序的输入来实现这一点。另外，中间认证机构可以代表其证书用户做出一个类似的决定。为了确保该证书用户正确执行这个要求，可以在策略约束扩展中设置 **inhibitPolicyMapping**。

8.1.4 认证通路处理

证书用户面临在两种策略之间做出选择：

a） 它可要求认证通路至少在用户预先确定的策略集中的一个策略下是有效的；或者

b） 它可要求通路验证模块报告认证通路为有效的策略集。

当证书用户知道时，第一种策略可能是最恰当的，因此，策略集对其计划的用途是可接受的。

当证书用户不知道时，第二种策略可能是最恰当的，因此，策略集对其计划的用途是可接受的。

在第一种情况下，认证通路验证程序将指明，只有在通路在 *initial-policy-set* 中规定的一个或多个策略下是有效的情况下，通路才是有效的，它将返回 *initial-policy-set* 的子集，在其下通路是有效的。在第二种情况下，认证通路验证程序可能指明，通路在 *initial-policy-set* 下是无效的，但在分离集 *authorities-constrained-policy-set* 下是有效的。而后证书用户将确定其计划的证书用途是否符合一个或多个证书策略，在这些策略下通路是有效的。通过将 *initial-policy-set* 设为 *any-policy*，如果在任何（未做规定）策略下通路都是有效的，那么证书用户可引起程序返回一个有效的结果。

8.1.5 自发放证书

有以下三种情况，在这三种情况下，认证机构可以向它自己发放证书：

a） 作为一种方便的公开密钥（它与用于签署证书的专用密钥相关）编码方法，因此它可以通过其使用证书的系统通信并保存为信任锚点；

b） 用于认证额外的 CA 公开密钥，CA 用于类别 a）所涵盖的目的之外的目的（如 OCSP，CRL 签署也有可能）；以及

c） 用于替换其自身已到期的各证书。

这些类型的证书称为自发放证书，它们可以通过以下事实得到认可，即出现在它们当中的发放者和对象名称是一样的。出于通路验证目的，类别 a）的自发放证书为自签署证书，因此利用包含在它们当中的公开密钥来验证，如果在通路中遇到它们，那么它们将被忽略。

类型 b）的自发放证书只能作为通路中的最终证书出现，并将作为最终证书进行处理。

类型 c）的自发放证书（也称为自发放中间证书）可以作为中间证书出现在通路中。作为一个良好的作法，在替换到期的密钥时，认证机构应请求发放任何带内交叉证书，它要求在使用密钥前替换公开密钥。然而，如果在通路中遇到此类别的自发放证书，那么将之作为中间证书来处理，以下除外：它们不为以下目的处理通路长度，即处理 **basicConstraints** 扩展的 **pathLenConstraint** 部件以及与 *policy-mapping-inhibitpending* and *explicit-policy-pending indicators* 相关的 *skip-certificates* 值。

如果机构使用相同的密钥来签署证书和 CRL，那么将使用一个单个的、类别 a）的自发放证书。如果机构使用不同的密钥来签署 CRL 而不是签署证书，那么机构可以选择发放两个类别 a）的自发放证书，每个对应一个密钥。在这种情况下，证书用户将需要访问两个自发放证书，以便为该机构签署的证书和 CRL 建立单独的信任锚点。可选地，机构可以为证书签署发放一个类别 a）的自发放证书，为 CRL 签署发放一个类别 b）的自发放证书。在这种情况下，证书用户将把在类别 a）的证书中经过认证的密钥，作为其有关该机构签署之证书和 CRL 的单个信任锚点。在这种情况下，如果使用类别 b）的自发放证书来验证 CRL 中的签名，那么在本标准中未定义任何可用于检查该证书有效性的方法。

如果在通路中遇到类别 b）的自发放证书，那么它们将被忽略。

注 — 分发 CA 公开密钥的其他机制不在本号码簿规范的讨论范围之内。

8.2 密钥和策略信息扩展

8.2.1 需求

以下需求与密钥和策略信息有关：

a） 可以定期地或在特殊情况下对 CA 密钥对进行更新。需要一个证书字段来传送公开密钥标识符，以供验证证书签名使用。在寻找正确的 CA 证书以确认证书发放者的公开密钥过程中，使用证书的系统可以使用此类标识符。

b） 通常，证书对象拥有不同的公开密钥，并且相应地，不同的证书对应不同的目的，例如，数字签名和加密密钥协议。需要一个证书字段来帮助证书用户为某个特定对象、出于某个特殊目的，选择正确的证书，或者允许 CA 规定，一个经过认证的密钥只能用于某个特殊的目的。

c） 可以定期地或在特殊情况下对对象密钥对进行更新。需要一个证书字段来传送公开密钥标识符，以区分在不同时间点上使用的、同一对象的不同公开密钥。在寻找正确的证书过程中，使用证书的系统可以使用此类标识符。

d） 典型地，对应已认证公开密钥的专用密钥用在不同于公开密钥有效期的时段中。利用数字签名密钥，典型地，签署专用密钥的用时比验证公开密钥的用时要短。证书的有效期用于指明公开密钥可以使用的期限，它不必等同于专用密钥的使用期限。在专用密钥遭到破坏的情况下，如果签名验证者知道专用密钥的有效使用期限，那么可以对泄漏期限做出限制。因此，要求能够指明证书中专用密钥的使用期限。

e） 由于证书可以用在可以使用多个证书策略的环境中，因此需要做出规定：应在证书中包括证书策略信息。

f） 当从一个组织机构到另一个组织机构进行交叉认证时，有时可以达成协定，认为两个组织机构中的某些策略是相同的。CA 证书需要允许证书发放者指明，它自身的其中一个证书策略等同于对象 CA 域中的另一个证书策略。这就是所谓的策略映射。

g） 使用本号码簿规范中定义之证书的加密或数字签名系统用户，需要能够事先确定其他用户支持的算法。

8.2.2 公开密钥证书和CRL扩展字段

定义了以下扩展字段：

a） 机构密钥标识符；

b） 对象密钥标识符；

c） 密钥用法；

d） 扩展的密钥用法；

e） 专用密钥使用期限；

f） 证书策略；

g） 策略映射。

这些扩展字段只能用作证书扩展，除了机构密钥标识符，它还可以用作 CRL 扩展。除非另行声明，否则这些扩展既可以用在 CA 证书中，也可以用在终端实体证书中。

8.2.2.1 机构密钥标识符扩展

本字段可当作一个证书扩展或 CRL 扩展来使用，用以确定用于验证本证书或 CRL 中签名的公开密钥。它使同一 CA 所用的不同密钥能够区别开来（例如，当发生密钥更新时）。本字段定义如下：

```
authorityKeyIdentifier EXTENSION ::= {
    SYNTAX              AuthorityKeyIdentifier
    IDENTIFIED BY       id-ce-authorityKeyIdentifier }

AuthorityKeyIdentifier ::= SEQUENCE {
    keyIdentifier               [0] KeyIdentifier               OPTIONAL,
    authorityCertIssuer         [1] GeneralNames                OPTIONAL,
    authorityCertSerialNumber   [2] CertificateSerialNumber     OPTIONAL }
    ( WITH COMPONENTS           {..., authorityCertIssuer PRESENT,
                            authorityCertSerialNumber PRESENT} |
    WITH COMPONENTS             {..., authorityCertIssuer ABSENT,
                            authorityCertSerialNumber ABSENT} )

KeyIdentifier ::= OCTET STRING
```

可以通过 **keyIdentifier** 部件中的一个明确的密钥标识符来确定密钥，通过鉴别有关密钥的证书（在 **authorityCertIssuer** 部件中给出证书发放者，在 **authorityCertSerialNumber** 部件中给出证书序列号），或者通过明确的密钥标识符和鉴别有关密钥的证书。如果两种鉴别形式都使用，那么证书或 CRL 发放者应确保它们是一致的。对发放机构，对证书或包含本扩展的 CRL，有关所有密钥标识符的密钥标识符应是惟一的。不要求支持本扩展的执行方案能够处理 **authorityCertIssuer** 部件中的所有名称形式。（有关 **GeneralNames** 类型的详细内容请参见第 8.3.2.1 节。）

认证机构将分配证书序列号，这样，每对（发放者、证书序列号）惟一确定一个单个的证书。**keyIdentifier** 形式可用于在构造通路期间选择 CA 证书。在构造通路期间，**authorityCertIssuer**、**authoritySerialNumber** 对只能用于为一个证书提供优先级。

本扩展总是为非关键的。

8.2.2.2 对象密钥标识符扩展

本字段用于确定正在进行认证的公开密钥。它使同一对象所用的不同密钥能够区别开来（例如，当发生密钥更新时）。本字段定义如下：

```
subjectKeyIdentifier EXTENSION ::= {
    SYNTAX              SubjectKeyIdentifier
    IDENTIFIED BY       id-ce-subjectKeyIdentifier }

SubjectKeyIdentifier ::= KeyIdentifier
```

对对象所用的所有密钥标识符而言，一个密钥标识符应是惟一的。本扩展总是为非关键的。

8.2.2.3 密钥用法扩展

本字段用于确定计划的用途，为这些用途发放了证书。对计划的用途可以通过策略做进一步约束。该策略可以在一个证书策略定义、一个合同或其他规范中予以声明。不过，策略不得取代由 **KeyUsage** 位指明的约束，例如，如果 **KeyUsage** 指明它只能用于密钥协议，那么证书策略不可以允许将证书用于数字签名。

在证书中设置一个特殊的 **KeyUsage** 值本身不能表明是一个通信实例，即通信各方依据本设置进行活动，例如，当签署一个文件时。方法定义，通过它各方可以表明其有关某个特殊通信实例的意图（例如，承诺有关该特殊实例的内容），已超出了本号码簿规范的讨论范围，但预计将存在多个方法。虽然不建议，但有可能使用证书内容，如证书策略，来指明签署的意图。不过，由于该声明是在 CA 发放证书之时做出的，因此此类用途可以不满足以下要求，即在签署者签署之时声明意图。

在 **keyUsage** 扩展的一个实例中可以对多个位进行设置。对多个位进行设置不得改变各单个位的含义，但应指明证书可用于各设置位所指明的所有目的。当对多个位进行设置时可能带来风险。在附件 I 中对这些风险进行了评论。

本字段定义如下：

```
keyUsage EXTENSION ::= {
    SYNTAX              KeyUsage
    IDENTIFIED BY       id-ce-keyUsage }

KeyUsage ::= BIT STRING {
    digitalSignature        (0),
    contentCommitment       (1),
    keyEncipherment         (2),
    dataEncipherment        (3),
    keyAgreement            (4),
    keyCertSign             (5),
    cRLSign                 (6),
    encipherOnly            (7),
    decipherOnly            (8) }
```

KeyUsage 类型中的各位如下所述：

a） **digitalSignature**：用于验证实体认证服务、数据源认证服务与/或完整性服务使用的数字签名。

b） **contentCommitment**：用于验证数字签名，旨在指明签署者承诺所签署的内容。对可用证书支持的承诺类型，可通过 CA 做进一步限制，例如，通过证书策略。签署者准确的承诺类型，例如，“评估和批准”或“计划绑定”，可以通过所签署的内容来指明，例如，已签署的文档本身或某些额外的已签署信息。

由于内容承诺签署认为是一个以数字形式签署的事务处理，因此 **digitalSignature** 位无需在证书中进行设置。如果设置了，它也不会影响签署者在已签署内容中所做的承诺水平。

注意：可以使用标识符 **nonRepudiation** 指向该 **keyUsage** 位。不过，已经反对使用该标识符。不管所用的标识符是什么，该位的语义都如本号码簿规范中所规定的那样。

c） **keyEncipherment**：用于加密密钥或其他安全信息，例如用于传达密钥。

d） **dataEncipherment**：用于加密用户数据，但不是密钥或上面 c）中的其他安全信息。

e） **keyAgreement**：当作一个公开密钥协议密钥使用。

f） **keyCertSign**：用于验证证书上的 CA 签名。

由于证书签署认为是 CA 对证书内容的一个承诺，因此对 **digitalSignature** 位和 **contentCommitment** 位都无需在证书中进行设置。如果设置了其中一个位（或两个位），它也不会影响签署者在已签署证书中所做的承诺水平。

g） **cRLSign**：用于验证 CRL 上的机构签名。

由于 CRL 签署认为是 CRL 发放者对 CRL 内容的一个承诺，因此对 **digitalSignature** 位和 **contentCommitment** 位都无需在证书中进行设置。如果设置了其中一个位（或两个位），它也不会影响签署者在已签署 CRL 中所做的承诺水平。

h） **encipherOnly**：当 **keyAgreement** 位也设置了时（指的是其他密钥用法位未定义），公开密钥协议密钥只能用于加密数据。

i） **decipherOnly**：当 **keyAgreement** 位也设置了时（指的是其他密钥用法位未定义），公开密钥协议密钥只能用于解密数据。

应用程序规范应指明，**digitalSignature** 或 **contentCommitment** 位中哪些适于其使用。如果签署应用程序不了解签署者有关内容承诺的意图，那么应用程序进行签署，并通过以下证书来为该签署提供支持，即该证书在其 **keyUsage** 扩展中设置了 **digitalSignature** 位。

即使利用证书（它只设置了 **digitalSignature** 位）对数字签名进行了验证，数字签名验证之外的其他因素也可以在确定签署意图方面发挥作用。相反地，即使利用证书（它只设置了 **contentCommitment** 位）对数字签名进行了验证，签署者也可以利用外部因素来取消对所签署内容的承诺。

keyCertSign 位只能用在 CA 证书中。如果 **KeyUsage** 设为 **keyCertSign**，那么 **basicConstraints** 扩展 cA 部件的值将设为 **TRUE**。CA 还可以使用 **KeyUsage** 中的其他已定义密钥用途位，例如，**digitalSignature** 位，它用于提供鉴权，以及在线管理事务处理的完整性。

可以由证书发放者选择决定，本扩展可以是关键的或者是非关键的。

如果扩展标志为关键的，或者如果扩展标志为非关键的但使用证书的系统认可它，那么证书只能用于以下目的，即对应的密钥用法位设为 1。如果扩展标志为非关键的，并且使用证书的系统不认可它，那么本扩展将被忽略。位设为 0 表明密钥不是供此目的使用。如果扩展的所有位都设为 0，那么密钥供上面所列目的之外的其他目的使用。

8.2.2.4 扩展的密钥用法扩展

除了在密钥用法扩展字段中所指明的基本目的之外，或者替代在密钥用法扩展字段中所指明的基本目的，本字段用于指明一个或多个其他目的，出于这些目的，可以使用经过认证的公开密钥。本字段定义如下：

```
extKeyUsage EXTENSION ::= {
    SYNTAX              SEQUENCE SIZE (1..MAX) OF KeyPurposeId
    IDENTIFIED BY       id-ce-extKeyUsage }

KeyPurposeId ::= OBJECT IDENTIFIER
```

通过利用 **anyExtendedKeyUsage** 标识符，CA 可以声明任何扩展的密钥用法。这使得 CA 可以发放一个包含有关扩展密钥用法的 OIDs 的证书，使用应用的证书可以提出这方面的要求，并且不限制证书只能用于这些密钥用法。如果扩展的密钥用法限制密钥用法，那么可以通过纳入该 OID 来移去这些限制。

```
anyExtendedKeyUsage  OBJECT IDENTIFIER  ::= { 2 5 29 37 0 }[S9]
```

密钥目的可以由任何组织机构以需求的形式进行定义。将依据 ITU-T X.660 建议书 | ISO/IEC 9834-1 来分配用于确定密钥目的的对象标识符。

本扩展可以是关键的，或者是非关键的，这由证书发放者决定。

如果扩展标志为关键的，那么证书只能用于其中一个所指明的目的。

如果扩展标志为非关键的，那么它指明密钥的计划目的，并可用于寻找实体（拥有多个密钥/证书）正确的密钥/证书。如果本扩展出现，那么使用证书的系统认可并处理 **extendedKeyUsage** 扩展类型，而后使用证书的系统应确保只将证书用于所指明的目的之一。（不过，使用应用程序可能需要指明一个特殊的目的，以便该应用程序接受证书。）

如果证书既包含关键密钥用法字段，又包含关键扩展密钥用法字段，那么将独立地对两个字段进行处理，并且证书只能用于与两个字段都相一致的目的。如果不存在任何与两个字段相一致的目的，那么证书不得用于任何目的。

本规范定义了以下关键目的，它们可以包括在扩展密钥用法扩展中。还可以包括在其他规范中定义的其他目的，如 IETF RFC 3280。

```
keyPurposes                    OBJECT IDENTIFIER  ::= {ds 38 1}
```

8.2.2.5 专用密钥使用期限扩展

本字段用于指明对应经过认证的公开密钥的专用密钥的使用期限。它只适用于数字签名密钥。本字段定义如下：

```
privateKeyUsagePeriod EXTENSION ::= {
    SYNTAX                PrivateKeyUsagePeriod
    IDENTIFIED BY         id-ce-privateKeyUsagePeriod }

PrivateKeyUsagePeriod ::= SEQUENCE {
    notBefore   [0]    GeneralizedTime OPTIONAL,
    notAfter    [1]    GeneralizedTime OPTIONAL }
    ( WITH COMPONENTS {..., notBefore PRESENT} |
    WITH COMPONENTS       {..., notAfter PRESENT} )
```

notBefore 部件用于指明专用密钥可以用于签署的最早日期和时间。如果 **notBefore** 部件未出现，那么未提供任何有关何时开始有效使用专用密钥的信息。**notAfter** 部件用于指明专用密钥可以用于签署的最晚日期和时间。如果 **notAfter** 部件未出现，那么未提供任何有关何时可以结束有效使用专用密钥的信息。

本扩展总是为非关键的。

注 1 — 专用密钥的有效使用周期可以不同于证书有效期所指明的公开密钥认证有效性。利用数字签名密钥，典型地，签署专用密钥的使用周期要短于验证公开密钥的使用周期。

注 2 — 如果数字签名的验证者想检查证书是否未被撤消，例如，因验证之时密钥折衷，那么在验证之时，有关公开密钥的有效证书将继续存在。在公开密钥证书到期后，签名验证者不能依赖通过 CRL 告知的折衷。

8.2.2.6 证书策略扩展

本字段列出了发放者 CA 认可的、适用于证书的各证书策略，以及与这些证书策略有关的、可选的限定符信息。证书策略清单用于确定证书通路的有效性，如第 10 节所述。在认证通路处理程序中不使用可选的限定符，但相关的限定符可以作为证书处理的输出提供，证书通过应用程序来帮助确定一个有效的通路对特定的事务处理是否合适。典型地，不同的证书策略将相关于不同的、可以使用经认证密钥的应用程序。在终端实体证书中出现本扩展将表明，有关该证书的证书策略是有效的。在由一个 CA 向另一个 CA 发放的证书中出现本扩展将表明，有关包含该证书的认证通路的证书策略可能是有效的。本字段定义如下：

```
certificatePolicies EXTENSION ::= {
    SYNTAX                CertificatePoliciesSyntax
    IDENTIFIED BY         id-ce-certificatePolicies }

CertificatePoliciesSyntax ::= SEQUENCE SIZE (1..MAX) OF PolicyInformation

PolicyInformation ::= SEQUENCE {
    policyIdentifier      CertPolicyId,
    policyQualifiers      SEQUENCE SIZE (1..MAX) OF
                              PolicyQualifierInfo OPTIONAL }

CertPolicyId ::= OBJECT IDENTIFIER
```

```
PolicyQualifierInfo ::= SEQUENCE {
    policyQualifierId       CERT-POLICY-QUALIFIER.&id
                            ({SupportedPolicyQualifiers}),
    qualifier               CERT-POLICY-QUALIFIER.&Qualifier
                            ({SupportedPolicyQualifiers}{@policyQualifierId})
                            OPTIONAL }

SupportedPolicyQualifiers CERT-POLICY-QUALIFIER ::= { ... }
```

PolicyInformation 类型的值用于确定和传达有关某个证书策略的限定符信息。部件 **policyIdentifier** 包含一个证书策略的标识符，部件 **policyQualifiers** 包含该元素的策略限定符值。

本扩展可以是关键的，或者是非关键的，这由证书发放者决定。

如果扩展标志为关键的，那么它指明证书只能用于本目的，依据的是其中一个已指明证书策略所隐含的规则。某个特殊策略的规则可以要求使用证书的系统对以某种特殊方式出现的限定符值进行处理。

如果扩展标志为非关键的，那么使用本扩展将无需限制对所列的策略使用证书。不过，证书用户可以要求出现某个特殊策略，以便使用证书（请参见第 10 节）。可以由证书用户来选择决定是处理还是忽略策略限定符。

证书策略和证书策略限定符类型可以由任何有需求的组织机构来定义。用于确定证书策略和证书策略限定符类型的对象标识符将依据 ITU-T X.660 建议书 | ISO/IEC 9834-1 进行指派。CA 可以利用 **anyPolicy** 标识符来声明 *any-policy*，以便将证书托付给所有可能的策略。由于需要确定该特殊值，以便应用，而不管应用或环境是什么，因此在本规范中指派该对象标识符。在本规范中将不为特殊的证书策略指派任何对象标识符。由定义证书策略的实体负责完成该指派任务。

```
anyPolicy               OBJECT IDENTIFIER  ::=          { 2 5 29 32 0 }
```

标识符 **anyPolicy** 不应有任何相关的策略限定符。

在定义证书策略限定符类型时，使用以下 ASN.1 对象类别：

```
CERT-POLICY-QUALIFIER ::= CLASS {
    &id                 OBJECT IDENTIFIER UNIQUE,
    &Qualifier          OPTIONAL }
WITH SYNTAX {
    POLICY-QUALIFIER-ID    &id
    [QUALIFIER-TYPE        &Qualifier] }
```

一个策略限定符类型的定义应包括：

— 可能的值的语义的声明；以及

— 指明限定符标识符是否可以出现在没有陪伴值的证书策略扩展中，以及如果出现，指明在这种情况下暗指的语义。

注 — 可以规定一个限定符可以拥有任何 ASN.1 类型。当预计将主要由没有 ASN.1 编码功能的应用来使用限定符时，建议应规定类型 **OCTET STRING**。那么 ASN.1 **OCTET STRING** 值可用于传达一个限定符值，依据策略元素定义机构所规定的某种约定来对限定符值进行编码。

8.2.2.7 策略映射扩展

本字段只能用在 CA 证书中，它允许证书发放者指明，出于包含本证书的认证通路用户的目的，可以认为其中的一个发放者证书策略等同于一个在对象 CA 域中所用的不同的证书策略。本字段定义如下：

```
policyMappings EXTENSION ::= {
    SYNTAX                  PolicyMappingsSyntax
    IDENTIFIED BY           id-ce-policyMappings }

PolicyMappingsSyntax ::= SEQUENCE SIZE (1..MAX) OF SEQUENCE {
    issuerDomainPolicy      CertPolicyId,
    subjectDomainPolicy     CertPolicyId }
```

issuerDomainPolicy 部件用于指明一个证书策略，它在发放 CA 的域中被认可，并可认为等同于 **subjectDomainPolicy** 部件中所指明的证书策略（它在对象 CA 的域中被认可）。

策略不得映射至特殊的值 **anypolicy** 或者自特殊的值 **anypolicy** 映射。

本扩展可以由证书发放者选择决定是关键的还是非关键的。建议它为关键的，否则证书用户不能正确解释对发放 CA 的约束。

注 1 — 策略映射的一个例子如下所述。美国政府域可以拥有一个称为加拿大贸易的策略，加拿大政府可以拥有一个称为美国贸易的策略。两个策略是分开确定和定义的，但两国政府之间可以存在一个协议，来接受出于相关目的而由这些策略暗指的规则中的跨边界认证通路。

注 2 — 策略映射意味着需要巨大的管理费用，以及在相关的决策中将涉及相称的、勤奋工作的、经过授权的人员。通常更倾向于就公共策略的更普遍使用达成协议，而不是应用策略映射。在上面的例子中，它将更倾向于美国、加拿大和墨西哥为北美贸易商定一个公共策略。

注 3 — 预计策略映射将只适用于有限的环境，这些环境中的策略声明会非常简单。

8.3 对象和发放者信息扩展

8.3.1 需求

以下需求与证书对象和证书发放者属性有关：

a） 证书需能供采用各种不同名称形式的应用使用，包括国际互联网电子邮件名称、国际互联网域名、X.400 发起者/接收者地址以及 EDI 方名称。因此，需要能够安全地将各种不同名称形式的多个名称与一个证书对象或者一个证书或 CRL 发放者关联起来。

b） 证书用户可能需要可靠地了解某些有关对象的鉴别信息，以便确信对象确实就是预期的人或物。例如，可能需要诸如邮政地址、在公司中的职位或图像之类的信息。此类信息可以方便地表示为号码簿属性，但这些属性不必是不同名称的一部分。因此除了那些在不同名称中的字段之外，还需要一个证书字段来传达额外的号码簿属性。

8.3.2 证书和CRL扩展字段

定义了以下扩展字段：

a） 对象可选的名称；

b） 发布者可选的名称；

c） 对象号码簿属性。

这些字段只能用作证书扩展，除了发布者可选的名称，它还可以用作 CRL 扩展。作为证书扩展，它们可以出现在 CA 证书中，或者出现在终端实体证书中。

8.3.2.1 对象可选的名称扩展

本字段包含一个或多个可选的名称，对由 CA 绑定于经认证的公开密钥的实体，可以使用众多名称形式中的任何一种。本字段定义如下：

```
subjectAltName EXTENSION ::= {
    SYNTAX              GeneralNames
    IDENTIFIED BY       id-ce-subjectAltName }

GeneralNames ::= SEQUENCE SIZE (1..MAX) OF GeneralName

GeneralName ::= CHOICE {
    otherName                   [0]     INSTANCE OF OTHER-NAME,
    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 }

OTHER-NAME ::= TYPE-IDENTIFIER

EDIPartyName ::= SEQUENCE {
    nameAssigner        [0]     DirectoryString {ub-name} OPTIONAL,
    partyName           [1]     DirectoryString {ub-name} }
```

GeneralName 类型可选值中的值为如下所示的不同形式的名称：

- **otherName** 为任何形式的名称，定义为 **OTHER-NAME** 信息对象类别的一个实例；
- **rfc822Name** 是一个国际互联网电子邮件地址，依据国际互联网 RFC 822 进行定义；
- **dNSName** 是一个国际互联网域名，依据国际互联网 RFC 1035 进行定义；
- **x400Address** 是一个 O/R 地址，依据 ITU-T X.411 建议书 | ISO/IEC 10021-4 进行定义；
- **directoryName** 是一个号码簿名称，依据 ITU-T X.501 建议书 | ISO/IEC 9594-2 进行定义；
- **ediPartyName** 是一个形式名称，在进行电子数据交换通信的合作伙伴之间商定；**nameAssigner** 部件用于确定一个机构，它在 **partyName** 部件中指派惟一的名称值；
- **uniformResourceIdentifier** 是一个万维网（WWW）统一资源标识符（URI），依据国际互联网 RFC 1630 进行定义；
- **iPAddress** 是一个网际协议（IP）地址，依据国际互联网 RFC 791 进行定义，以二进制字符串形式表示；
- **registeredID** 是一个任何已注册对象的标识符，依据 ITU-T X.660 建议书 | ISO/IEC 9834-1 进行指派。

对在 **GeneralName** 类型中所用的每种名称形式，都有一个名称注册系统来确保所用的任何名称能够明确地从两个证书发放者和证书用户当中鉴别出一个实体来。

由证书发放者来决定，本扩展是关键的还是非关键的。不要求支持本扩展的执行方案能够处理所有的名称形式。如果扩展标志为关键的，那么应认可和处理至少一个出现的名称形式，否则认为证书是无效的。除了前面所述的限制，允许使用证书的系统忽略任何带不认可或不支持名称形式的名称。如果证书的对象字段包含一个号码簿名称，它可明确确定对象，那么建议将本字段标志为非关键的。

注 1 — 关于 **TYPE-IDENTIFIER** 类别的使用，在 ITU-T X.681 建议书 | ISO/IEC 8824-2 的附件 A 和附件 C 中予以描述。

注 2 — 如果本扩展字段出现，并标志为是关键的，那么证书的 **subject** 字段可以包含一个空名称（例如，一个零相对不同的名称序列），在这种情况下，只能通过本扩展中的名称来确定对象。

8.3.2.2 发放者可选的名称扩展

本字段包含一个或多个可选的名称，对证书或 CRL 发放者，可以使用众多名称形式中的任何一种。本字段定义如下：

```
issuerAltName EXTENSION ::= {
    SYNTAX              GeneralNames
    IDENTIFIED BY       id-ce-issuerAltName }
```

由证书或 CRL 发放者来决定，本扩展是关键的还是非关键的。不要求支持本扩展的执行方案能够处理所有的名称形式。如果扩展标志为关键的，那么应认可和处理至少一个出现的名称形式，否则认为证书或 CRL 是无效的。除了前面所述的限制，允许证书使用系统忽略任何带不认可或不支持名称形式的名称。如果证书或 CRL 的发放者字段包含一个号码簿名称，它可明确确定发放机构，那么建议将本字段标志为非关键的。

注 — 如果本扩展字段出现，并标志为关键，那么证书的 **issuer** 字段或 CRL 可以包含一个空名称（例如，一个 0 相对不同名称序列），在这种情况下，只能通过本扩展中的名称来确定发放者。

8.3.2.3 对象号码簿属性扩展

本字段为证书的对象传达任何期望的号码簿属性值。本字段定义如下：

```
subjectDirectoryAttributes EXTENSION ::= {
    SYNTAX              AttributesSyntax
    IDENTIFIED BY       id-ce-subjectDirectoryAttributes }

AttributesSyntax ::= SEQUENCE SIZE (1..MAX) OF Attribute
```

由证书发放者来决定，本扩展是关键的还是非关键的。不要求使用系统处理本扩展的证书能够理解包括在本扩展中的所有属性类型。如果扩展标志为关键的，那么应理解至少一个包含在本扩展中的属性类型，以便接受证书。如果扩展标志为关键的，并且任何一个包含的属性类型都不被理解，那么证书将被拒绝。

如果本扩展出现在公开密钥证书中，那么在第 15 节中定义的某些扩展也可以出现。

8.4 认证通路约束扩展

8.4.1 需求

对认证通路处理：

a） 终端实体证书需可区别于 CA 证书，以便防止终端实体将其自身建成为无需认证的 CA。还需要使 CA 有可能对后续链（来自经过认证的对象 CA）的长度做出限制，例如，不大于一个证书或不大于两个证书。

b） CA 需要能够规定约束，这将使证书用户能够检查认证通路中的低信任 CA（即认证通路比 CA 更低的 CA，证书用户以其公开密钥开始）是否未与其信任发生冲突（冲突因在一个不合适的名称空间中向对象发放证书而发生）。认证用户应能够自动检查这些约束条件是否得到了遵守。

c） 需要在自动的、自包含的模块中能够执行认证通路处理任务。这需要允许执行可信的硬件或软件模块，由这些模块来执行认证通路处理功能。

d） 无需依靠与本地用户的实时交互，也应有可能执行认证通路处理任务。

e） 无需依靠使用有关策略描述信息的、可信的本地数据库，也应有可能执行认证通路处理任务。（为了执行认证通路处理任务，至少需要一些可信的本地信息 — 最初的公开密钥，但此类信息的量应尽可能少。）

f） 认证通路需工作于以下环境，即当中认可多个证书策略。CA 需要能够规定它信任哪些其他域中的 CA，以及出于什么目的。需要支持通过多个策略域的链接。

g） 要求信任模型具备彻底的灵活性。当考虑多个互连企业的需求时，严格的层次结构模型是不合适的，它对单个的组织机构合适。在选择认证通路中的第一个可信 CA 中要求具备灵活性。尤其是，它应有可能要求认证通路开始于公开密钥用户系统的本地安全域中。

h） 命名结构不应受到在证书中使用名称之需求的约束，即无需为了满足认证机构的要求而对组织机构或地理区域认为是正常的号码簿名称结构进行调整。

i） 证书扩展字段需要向后兼容于未受约束的认证通路处理系统，如较早版本的 ITU-T X.509 建议书 | ISO/IEC 9594-8 所规定的那样。

j） CA 需要能够禁止使用策略映射，并要求在认证通路的后续证书中出现显性证书策略标识符。

注 — 在任何使用证书的系统中，认证通路的处理都需要相当的担保水准。本号码簿规范定义了可能在执行方案中用到的功能，执行方案需要符合特定的担保声明。例如，担保要求可能声明，应保证认证通路处理的处理工作免遭破坏（例如，篡改软件或修改数据）。担保水准应与商业风险相称。例如：

— 对用于确认高额资金转账的公开密钥，可能需要对相应的密码模块进行内部处理；然而

— 对家庭银行余额查询，软件形式的处理可能就够了。

因此，对硬件密码模块形式（或者密码令牌作为一种可选形式）的执行方案，认证通路处理功能应是合适的。

k） CA 需要能够防止特殊值 any-policy 被认为是认证通路后续证书中的一个有效策略。

8.4.2 证书扩展字段

定义了以下扩展字段：

a） 基本的约束；

b） 名称约束；

c） 策略约束；

d） 禁止任何策略。

这些扩展字段只能用做证书扩展，名称约束和策略约束只能用在 CA 证书中；基本的约束还可以用在终端实体证书中。在附件 G 中给出了使用这些扩展的例子。

8.4.2.1 基本的约束扩展

本字段用于指明对象是否可以当做一个 CA，经认证的公开密钥用于确认证书签名。如果可以，那么还可以规定一个认证通路长度约束。本字段定义如下：

```
basicConstraints EXTENSION ::= {
    SYNTAX              BasicConstraintsSyntax
    IDENTIFIED BY       id-ce-basicConstraints }

BasicConstraintsSyntax ::= SEQUENCE {
    cA                      BOOLEAN DEFAULT FALSE,
    pathLenConstraint       INTEGER (0..MAX) OPTIONAL }
```

cA 部件用于指明经认证的公开密钥是否可以用于确认证书签名。

只有当 **cA** 设为 TRUE 时，**pathLenConstraint** 部件才出现。它给出了在一个认证通路中可以紧跟本证书之后的、CA 证书的最大数量。值 0 指明本证书对象只能向终端实体发放证书，而不能进一步向 CA 发放证书。如果没有任何 **pathLenConstraint** 字段出现在认证通路的任何证书中，那么对允许的认证通路长度没有限制。约束条件从通路中的下一个证书开始起作用。约束条件限制了包含本扩展的证书与终端实体证书之间的、认证通路片段的长度。它对信任锚点与包含本扩展的证书之间的、认证通路中的 CA 证书数量没有影响。因此，一个完整的、认证通路的长度可以超过受本扩展限制的、片段的最大长度。约束条件用于控制包含约束条件的 CA 证书与终端实体证书之间的、非自发放 CA 证书的数量。因此，本通路片段的总的长度，不包括自发放证书，可以比约束值多两个证书。（这包括片段两个端点上的证书，加上受本扩展值限制的、两个端点之间的 CA 证书。）

由证书发放者来决定，本扩展是关键的还是非关键的。建议它被标志为关键的，否则未授权为 CA 的实体可以发放证书，并且证书使用系统可以无意地使用这样一个证书。

如果本扩展字段出现并标志为关键，或者标志为非关键但被使用证书的系统所认可，那么：

— 如果 **cA** 的值未设为 TRUE，那么不得将经认证的公开密钥用于验证证书签名；

— 如果 **cA** 的值设为 TRUE，并且 **pathLenConstraint** 出现，那么使用证书的系统将检查正在处理的认证通路是否与 **pathLenConstraint** 的值相一致。

注 1 — 如果本扩展字段未出现，或者标志为非关键且不被使用证书的系统所认可，那么认为证书是一个终端实体证书，不能用于验证证书签名。

注 2 — 如果限定证书对象只能是一个终端实体，即不是一个 CA，那么发放者可以包括本扩展字段，而字段只能包含一个空的 **SEQUENCE** 值。

8.4.2.2 名称约束扩展

本字段只能在 CA 证书中使用，用于指明名称空间，必须在该名称空间内确定认证通路上后续证书中的所有对象名称。本字段定义如下：

```
nameConstraints EXTENSION ::= {
    SYNTAX          NameConstraintsSyntax
    IDENTIFIED BY       id-ce-nameConstraint }

NameConstraintsSyntax ::= SEQUENCE {
    permittedSubtrees           [0]     GeneralSubtrees OPTIONAL,
    excludedSubtrees            [1]     GeneralSubtrees OPTIONAL,
    requiredNameForms           [2]     NameForms OPTIONAL  }

GeneralSubtrees ::= SEQUENCE SIZE (1..MAX) OF GeneralSubtree
```

```
GeneralSubtree ::= SEQUENCE {
     base                 GeneralName,
     minimum    [0]    BaseDistance DEFAULT 0,
     maximum    [1]    BaseDistance OPTIONAL }

BaseDistance ::= INTEGER (0..MAX)

NameForms  ::= SEQUENCE {
    basicNameForms   [0]    BasicNameForms OPTIONAL,
    otherNameForms   [1]    SEQUENCE SIZE (1..MAX) OF OBJECT IDENTIFIER OPTIONAL }
(ALL EXCEPT ({ -- 无；即至少应出现一个部件。  -- }))


BasicNameForms  ::= BIT STRING {
     rfc822Name                 (0),
     dNSName                    (1),
     x400Address                (2),
     directoryName              (3),
     ediPartyName               (4),
     uniformResourceIdentifier  (5),
     iPAddress                  (6),
     registeredID               (7) }  (SIZE (1..MAX))
```

如果出现，那么 **permittedSubtrees** 和 **excludedSubtrees** 部件各规定一个或多个命名子树，通过子树的根名称来定义各个子树，并且可选地，在该子树中，这是一个通过上级与/或下级来限制的区域。如果 **permittedSubtrees** 出现，那么这些子树中的各对象名称是可接受的。如果出现，那么由对象 CA 或认证通路中后续 CA 发放的任何证书（拥有一个这些子树中的对象名称）都将是不可接受的。如果 **permittedSubtrees** 和 **excludedSubtrees** 都出现，并且名称空间出现重叠，那么对重叠区域内的名称，排除声明将优先。如果对某个名称形式未规定允许的子树，也未规定排除的子树，那么在名称形式中的任何名称都将是可接受的。如果 **requiredNameForms** 出现，那么认证通路中的所有后续证书都必须包括一个至少以一种要求之名称形式存在的名称。

如果 **permittedSubtrees** 出现，那么以下内容适用于通路中的所有后续证书。如果某个证书包含一个以下名称形式的对象名称（在 **subject** 字段或 **subjectAltNames** 扩展中），即允许子树为该名称形式而规定，那么其名称必须落在至少一个规定的子树内。如果某个证书只包含以下名称形式的对象名称，即用于规定允许子树的名称形式之外的名称形式，那么不要求对象名称落在任何规定的子树内。例如，假设规定了两个允许子树，一个用于 DN 名称形式，一个用于 rfc822 名称形式，未规定任何排除子树，但规定了 **directoryName** 位和 **rfc822Name** 位需出现的 **requiredNameForms**。只包含号码簿名称或 rfc822 名称之外的名称的证书将是不可接受的。不过，如果未规定 **requiredNameForms**，那么这样一个证书将是可接受的。例如，假设规定了两个允许子树，一个用于 DN 名称形式，一个用于 rfc822 名称形式，未规定任何排除子树，并且 **requiredNameForms** 未出现。只包含 DN、且 DN 在规定的允许子树内的证书将是可接受的。既包含 DN 又包含 rfc822 名称、且其中只有一个在其规定的允许子树内的证书将是不可接受的。只包含 DN 或 rfc822 名称之外的名称的证书也将是可接受的。

注 — 本例只有说明性作用。**directoryName** 名称形式除外，关于如何处理其层次结构中 **GeneralName** 类型名称形式的名称，不在本建议书 | 国际标准中做详细说明。

如果 **excludedSubtrees** 出现，那么任何由对象 CA 或认证通路中后续 CA 发放的证书（拥有一个这些子树中的对象名称，在 **subject** 字段或 **subjectAltNames** 扩展中）都将是不可接受的。例如，假设规定了两个排除子树，一个用于 DN 名称形式，一个用于 rfc822 名称形式。只包含 DN、且 DN 在规定的排除子树内的证书将是可接受的。既包含 DN 又包含 rfc822 名称、且其中至少有一个在其规定的排除子树内的证书将是不可接受的。

当证书对象拥有多个相同名称形式的多个名称时（包括在 **directoryName** 名称形式情况下、非空的、证书对象字段中的名称），那么将对所有此类名称进行测试，看它们是否满足有关该名称形式的名称约束要求。

如果 **requiredNameForms** 出现，那么认证通路中的所有后续证书都必须包括一个以至少一种要求的名称形式出现的对象名称。

对通过 **GeneralName** 类型可用的名称形式，只有那些拥有一个良好定义的层次结构的名称形式才可以用在 **permittedSubtrees** 和 **excludedSubtrees** 字段中。**directoryName** 名称形式满足此要求；当使用该名称形式时，一个命名子树对应一个 DIT 子树。

minimum 字段用于规定子树内区域的上界。其最终名称部件在所规定级之上的所有名称都不包含在该区域内。等于 0 的 **minimum** 值（缺省值）对应基础，即子树的顶层节点。例如，如果 **minimum** 设为 1，那么命名子树不包括基础节点，但包括从属节点。

maximum 字段用于规定子树内区域的下界。其最终名称部件在所规定级之下的所有名称都不包含在该区域内。等于 0 的 **maximum** 值对应基础，即子树的顶层。未出现 **maximum** 部件则表明不应对子树内的区域施加任何下级限制。例如，如果 **maximum** 设为 1，那么除了子树基础及其直接的从属，命名子树不包括任何节点。

对 **directoryName** 名称形式，如果 **RDN**s 的 **SEQUENCE**（它形成 **base** 中完整的 **DN**）等同于 **RDN**s 同一号码的最初 **SEQUENCE**（它形成 **certificate** 的 **subject** 字段中的第一部分 **DN**），那么认为 **certificate** 从属于 **base**（并因此成为子树中的候选者）。**certificate** 的 **subject** 字段中的 **DN** 在其序列中可以拥有额外的跟踪 **RDN**s，它们不出现在 **base** 的 **DN** 中。**distinguishedNameMatch** 匹配规则用于比较 **base** 的值和证书 **subject** 字段中 **DN** 中的 **RDNs** 最初序列。

由证书发放者来决定，本扩展是关键的还是非关键的。建议它被标志为关键的，否则证书用户不可以对认证通路中的后续证书进行检查，所述的认证通路位于发放 CA 方案的名称空间中。

不要求符合要求的执行方案认可所有可能的名称形式。

如果扩展出现，并被标志为关键的，那么使用证书的执行方案必须认可并处理所有的名称形式，对这些名称形式，在扩展中存在子树规范（允许或排除），并且在认证通路任何后续证书的 **subject** 字段或 **subjectAltNames** 扩展中都存在一个对应的值。如果在子树规范和后续证书中都出现了未被认可的名称形式，那么将按以下方式对证书进行处理，即仿佛遇到了一个未被认可的、关键的扩展。如果证书中的任何对象名称都落在了一个排除在外的子树中，那么证书是不可接受的。如果为未包括在任何后续证书中的名称形式规定了一个子树，那么将忽略该子树。如果 **requiredNameForms** 部件只规定了未被认可的名称形式，那么将按以下方式对证书进行处理，即仿佛遇到了一个未被认可的、关键的扩展。否则，在通路的所有后续证书中，必须至少出现一种被认可的名称形式。

如果扩展出现，并被标志为非关键的，并且使用证书的执行方案不认可在任何 **base** 部件中所用的名称形式，那么可以忽略子树规范。如果扩展被标志为非关键的，并且使用证书的执行方案不认可在 **requiredNameForms** 部件中所规定的任何名称形式，那么将按以下方式对证书进行处理，即仿佛 **requiredNameForms** 部件未出现。

注意：在某些情况下，可能需要从一个 CA 向另一个 CA 发放多个证书，以便在某些名称约束要求出现冲突的情况下获得要求的结果。例如，假设 Acme 公司在美国有 20 个分支机构。

Widget 公司想对 Acme 公司的中心 CA 进行交叉认证，但只想 Widget 团体将 Acme 证书用于满足以下准则的对象：

— Acme 公司的分支机构 1-19，所有部门都将当作是对象；

— Acme 公司的分支机构 20，除了销售部门中的对象，所有部门都将不当作是对象。

这可以通过发放两个如下所述的证书来实现；第一个证书将有一个为 {base: C=US, O=Acme} 的 **permittedSubtrees** 以及一个为 {base: C=US, O=Acme, OU=branch20} 的 **excludedSubtrees**。第二个证书将有一个为 {base: C=US, O=Acme, OU=branch20, OU=Purchasing} 的 **permittedSubtrees**。

附件 G 包含有关名称约束扩展如何使用的例子。

8.4.2.3 策略约束扩展

本字段用于规定以下约束，这些约束可以要求明确确定证书策略，或者禁止对认证通路的剩余部分进行策略映射。本字段定义如下：

```
policyConstraints EXTENSION ::= {
    SYNTAX              PolicyConstraintsSyntax
    IDENTIFIED BY       id-ce-policyConstraints }

PolicyConstraintsSyntax ::= SEQUENCE {
    requireExplicitPolicy     [0]     SkipCerts OPTIONAL,
    inhibitPolicyMapping      [1]     SkipCerts OPTIONAL }
```

```
SkipCerts ::= INTEGER (0..MAX)
```

如果 **requireExplicitPolicy** 部件出现，并且认证通路包括一个由已命名 CA 发放的证书，那么通路中的所有证书都需要在证书策略扩展中包含一个可接受的策略标识符。可接受的策略标识符是认证通路用户要求的认证策略标识符、策略（通过策略映射，已宣布该策略等同于这些策略中的其中一个策略）标识符，或者是 *any-policy*。已命名 CA 要么是包含本扩展的证书的发放者 CA（如果 **requireExplicitPolicy** 的值为 0），要么是作为认证通路中某个后续证书发放者的 CA（通过一个非 0 的值来指出）。

如果 **inhibitPolicyMapping** 部件出现，那么它指明，在从认证通路中一个已命名 CA 开始到认证通路终点结束的所有证书中，均不允许策略映射。已命名 CA 要么是包含本扩展的证书的对象 CA（如果 **inhibitPolicyMapping** 的值为 0），要么是作为认证通路中某个后续证书对象的 CA（通过一个非 0 的值来指出）。

类型 **SkipCerts** 的值用于指明认证通路中的证书数量，以便在约束生效之前跳过。

由证书发放者来决定，本扩展是关键的还是非关键的。建议它被标志为关键的，否则证书用户不能正确解释发放 CA 的约定。

8.4.2.4 禁止任何策略扩展

本字段用于规定一个约束，它指明，对开始于一个命名 CA 的认证通路中的所有非自放证书，不认为 *any-policy* 明确匹配于其他证书策略。命名 CA 可以是包含本扩展的证书的对象 CA（如果 **inhibitAnyPolicy** 的值为 0），或者是这样一个 CA，即它是认证通路中后续证书的对象（通过一个非 0 的值来指明）。

```
inhibitAnyPolicy    EXTENSION       ::= {
    SYNTAX          SkipCerts
IDENTIFIED BY       id-ce-inhibitAnyPolicy }
```

由证书发放者来决定，本扩展是关键的还是非关键的。建议它被标志为关键的，否则证书用户不能正确解释发放 CA 的约定。

8.5 基本的CRL扩展

8.5.1 需求

以下需求与 CRL 相关：

a） 证书用户需要能够跟踪由 CRL 发放者或 CRL 分发点发放的所有 CRL（请参见第 8.6 节），并且能够检测序列中丢失的 CRL。因此需要 CRL 序列号。

b） 某些 CRL 用户可能希望对撤消做出不同的响应，这依赖于撤消的理由。因此，要求 CRL 条目能够指明撤消的理由。

c） 要求机构能够暂停某个证书的有效性，并且能够在之后撤消它或恢复它。对这样一个行动，可能的理由包括：

— 当撤消请求未得到证实并且没有充分的信息来确定它是否有效时，希望减少错误撤消的倾向；

— 其他的商业需求，如在审计或调查期间暂停某个实体的证书。

d） 为每个已撤消证书，CRL 都包含一个机构执行撤消的日期。可以知道的更多信息有 — 一个真实的或受到怀疑的密钥破坏何时发生，该信息对证书用户是有用的。撤消日期不足以解决某些争议，原因是，做最坏的假设，所有在证书有效期发放的证书都将被认为是无效的。不过，以下这点对用户可能是重要的 — 签名产生之后，即使用于签署信息的密钥遭到了破坏，也应认为经过签署的文件是有效的。为有助于该问题的解决，CRL 条目应包括第二个日期，它用于指明何时知道或怀疑专用密钥遭到了破坏。

e） 证书用户需要能够从 CRL 自身确定额外的信息，包括本清单所涵盖证书的范围、撤消通知的排序，以及哪个 CRL 流的 CRL 号码在其中是惟一的。

f） 发放者需要能够动态改变 CRL 的分割，并且如果分割发生变化，能够将证书用户指向新的、相关 CRL 的位置。

g） Delta CRL 还可用于更新某个特定的基础 CRL。证书用户需要能够从某个特定的 CRL 确定 delta CRL 是否可用、它们在哪里、何时发放下一个 delta CRL。

h） CRL除了发布通知告知证书已经撤消之外，还需发布通知告知将在未来某个规定的日期和时间撤消证书。

i） 需要提供更有效的方式来指明，已从 CRL 中撤消一系列证书。

8.5.2 CRL和CRL条目扩展字段

定义了以下扩展字段：

a） *CRL 号码*；

b） *理由代码*；

c） *持有指令代码*；

d） *无效日期*；

e） *CRL 范围*；

f） *状态命名*；

g） *CRL 流标识符*；

h） *排好序的清单*；

i） *Delta 信息*。

CRL 号码、CRL 范围、状态命名、CRL 流标识符、排好序的清单和 delta 信息只能用作 CRL 扩展字段，其他的字段只能用作 CRL 条目扩展字段。

8.5.2.1 CRL号码扩展

本 CRL 扩展字段为每个由某个特定 CRL 发放者通过某个特定机构号码簿属性或 CRL 分发点发放的 CRL 传送一个单调增长的序列号。它允许 CRL 用户对在所处理 CRL 之前发放的 CRL 进行检测，看其是否也已经过检测和处理。本字段定义如下：

```
cRLNumber EXTENSION ::= {
    SYNTAX              CRLNumber
    IDENTIFIED BY       id-ce-cRLNumber }

CRLNumber ::= INTEGER (0..MAX)
```

本扩展总是为非关键的。

8.5.2.2 理由代码扩展

本 CRL 条目扩展字段为证书撤消确定一个理由。应用可以基于本地策略，利用理由代码来决定如何对提出的撤消做出响应。本字段定义如下：

```
reasonCode EXTENSION ::= {
    SYNTAX              CRLReason
    IDENTIFIED BY       id-ce-reasonCode }

CRLReason ::= ENUMERATED {
    unspecified             (0),
    keyCompromise           (1),
    cACompromise            (2),
    affiliationChanged      (3),
    superseded              (4),
    cessationOfOperation    (5),
    certificateHold         (6),
    removeFromCRL           (8),
    privilegeWithdrawn      (9),
    aaCompromise            (10) }
```

以下理由代码值指明为什么撤消一个证书：

— **unspecified** 可用于撤消证书，出于特定代码之外的理由；

— **keyCompromise** 用于撤消终端实体证书；它指明已经知道或怀疑对象的专用密钥或证书中所验证之对象的其他方面遭到了破坏；

— **cACompromise** 用于撤消 CA 证书；它指明已经知道或怀疑对象的专用密钥或证书中所验证之对象的其他方面遭到了破坏；

— **affiliationChanged** 用于指明证书中的对象名称或其他信息已被修改，但没有理由怀疑专用密钥遭到了破坏；

— **superseded** 用于指明证书已被替换，但没有理由怀疑专用密钥遭到了破坏；

— **cessationOfOperation** 用于指明其发布的目的已不再需要证书，但没有理由怀疑专用密钥遭到了破坏；

— **privilegeWithdrawn** 用于指明一个证书（公开密钥证书或属性证书）已被撤消，原因是，包含在该证书中的特权已被撤消；

— **aACompromise** 用于指明，已经知道或怀疑属性证书中所验证之 AA 的某些方面遭到了破坏。

可以通过发放一个带理由代码 **certificateHold** 的 CRL 来持有一个证书。证书持有通知可以包括一个可选的持有指令代码，以便向证书用户传达额外的信息（请参见第 8.5.2.3 节）。一旦持有，那么可以用以下三种方法中的一种来对它进行处理：

a） 它可以保留在 CRL 中，而不采取进一步行动，致使用户拒绝在持有期间发放的事务处理；或者

b） 可以通过（最终）撤消相同的证书来替换它，在这种情况下，理由应为标准的撤消理由之一，撤消日期应为持有证书的日期，并且不得出现可选的指令代码扩展字段；或者

c） 可以明确释放它，并从 CRL 中删去条目。

removeFromCRL 理由代码只能供 delta-CRL（请参见第 8.6 节）使用，并指明，由于证书到期或持有释放，目前应删去一个现有的 CRL 条目。带该理由代码的条目将用在 delta-CRL 中，其对应的基础 CRL 或任何后续的 CRL（delta 或对范围而言是完整的）包含一个有关同一证书（带理由代码 **certificateHold**）的条目。

本扩展总是为非关键的。

8.5.2.3 持有指令代码扩展

本 CRL 条目扩展字段包括一个已注册的指令标识符，用于指明当遇到一个已有的证书时需采取什么行动。它只在拥有一个 certificateHold 理由代码的条目中适用。本字段定义如下：

```
holdInstructionCode EXTENSION ::= {
    SYNTAX              HoldInstruction
    IDENTIFIED BY       id-ce-instructionCode }

HoldInstruction ::= OBJECT IDENTIFIER
```

本扩展总是为非关键的。在本号码簿规范中未定义任何标准的持有指令代码。

注 — 持有指令的例子可以是“请与 CA 通信”或者“收回用户的令牌”。

8.5.2.4 无效日期扩展

本 CRL 条目扩展字段用于指明知道或怀疑专用密钥受到危害或者否则认为证书无效的日期。该日期可以早于 CRL 条目中的撤消日期，它是机构处理撤消的日期。本字段定义如下：

```
invalidityDate EXTENSION ::= {
    SYNTAX              GeneralizedTime
    IDENTIFIED BY       id-ce-invalidityDate }
```

本扩展总是为非关键的。

注 1 — 本扩展中的日期本身对不可否认目的是不够的。例如，该日期可能是一个专用密钥持有者建议的日期，此人可能欺骗性地声称密钥在过去的某个时候遭到了破坏，以便否认一个有效产生的签名。

注 2 — 当撤消首先由 CRL 中的某个机构提出时，无效日期可能在早期 CRL 的发放日期之前。撤消日期不得在早期 CRL 的发放日期之前。

8.5.2.5 CRL范围扩展

注 — 不赞成使用 CRL 范围扩展。

批注 [S143]: Page: 37
Fix for DR 284 published in TC 2

在使用以下 CRL 扩展的 CRL 中指明 CRL 的范围。为了防止对不支持范围扩展的应用造成替换攻击，如果出现，范围扩展将标志为关键的。

本扩展可用于提供有关各种不同 CRL 类型的范围声明，包括：

— 简单的 CRL，提供有关证书（由一个单个机构发放）的撤消信息；

— 间接的 CRL，提供有关证书（由多个机构发放）的撤消信息；

— delta-CRL，更新先前发布的撤消信息；

— 间接的 delta-CRL，提供撤消信息，用于更新多个基础 CRL（由一个单个或多个机构发放）。

```
crlScope EXTENSION ::= {
    SYNTAX                  CRLScopeSyntax
    IDENTIFIED BY           id-ce-cRLScope }

CRLScopeSyntax ::=          SEQUENCE SIZE (1..MAX) OF PerAuthorityScope

PerAuthorityScope ::= SEQUENCE {
    authorityName                   [0]     GeneralName OPTIONAL,
    distributionPoint               [1]     DistributionPointName OPTIONAL,
    onlyContains                    [2]     OnlyCertificateTypes OPTIONAL,
    onlySomeReasons                 [4]     ReasonFlags OPTIONAL,
    serialNumberRange               [5]     NumberRange OPTIONAL,
    subjectKeyIdRange               [6]     NumberRange OPTIONAL,
    nameSubtrees                    [7]     GeneralNames OPTIONAL,
    baseRevocationInfo              [9]     BaseRevocationInfo OPTIONAL
    }

OnlyCertificateTypes ::= BIT STRING {
    user            (0),
    authority       (1),
    attribute       (2) }

NumberRange ::= SEQUENCE {
    startingNumber          [0]     INTEGER OPTIONAL,
    endingNumber            [1]     INTEGER OPTIONAL,
    modulus                         INTEGER OPTIONAL }

BaseRevocationInfo ::= SEQUENCE {
    cRLStreamIdentifier         [0]     CRLStreamIdentifier OPTIONAL,
    cRLNumber                   [1]     CRLNumber,
    baseThisUpdate              [2]     GeneralizedTime }
```

如果 CRL 是一个间接的 CRL，它为多个机构提供撤消状态信息，那么扩展将包括多个 **PerAuthorityScope** 构件，每个机构对应一个或多个构件，为它纳入撤消信息。与机构（非发放本 CRL 的机构）相关的每个 **PerAuthorityScope** 实例都将包含 **authorityName** 部件。如果 CRL 是一个 dCRL，它为多个基础 CRL（由一个单个的机构发放）提供 delta 撤消状态信息，那么扩展将包括多个 **PerAuthorityScope** 构件，每个基础 dCRL 对应一个构件，本 dCRL 为它提供更新。即使有多个 **PerAuthorityScope** 构件实例，如果出现，对所有实例，**authorityName** 部件的值都将是相同的。

如果 CRL 是一个间接 dCRL，它为由多个机构发放的多个基础 CRL 提供 delta 撤消状态信息，那么扩展将包括多个 **PerAuthorityScope** 构件，每个对应一个基础 CRL，该 dCRL 为这些 CRL 提供更新。与机构而不是本间接 dCRL 发放者相关的每个 **PerAuthorityScope** 实例，都将包括 **authorityName** 部件。

对扩展中出现的各 **PerAuthorityScope** 实例，按以下方式来使用各字段。注意：在间接 CRL 和间接 dCRL 情况下，各 **PerAuthorityScope** 实例可以包含这些字段的不同组合和不同值。

如果 **authorityName** 字段出现，那么确定发放证书的机构（为其提供撤消信息）。如果省略 **authorityName**，那么它缺省为 CRL 发放者名称。

如果 **distributionPoint** 字段出现，那么按 **issuingDistributionPoint** 扩展中所述的方式进行使用。

如果 **onlyContains** 字段出现，那么指明 CRL 包含撤消状态信息的证书类型。如果本字段未出现，那么 CRL 包含有关所有证书类型的信息。

如果 **onlySomeReasons** 字段出现，那么按 **issuingDistributionPoint** 扩展中所述的方式进行使用。

如果 **serialNumberRange** 元素出现，那么按以下方式进行使用。当一个模值出现时，在检查是否出现在范围内之前，将以给定值为模对序列号进行缩减。而后，认为带（经过缩减的）序列号的证书处于 CRL 范围内，如果它：

— 等于或大于 **startingNumber**，并小于 **endingNumber**，二者都出现；或者

— 等于或大于 **startingNumber**，当 **endingNumber** 不出现时；或者

— 小于 **endingNumber**，当 **startingNumber** 不出现时。

如果出现，认为 **subjectKeyIdRange** 元素等同于 **serialNumberRange**，除非所用的号码为证书 **subjectKeyIdentifier** 扩展中的值。**BIT STRING** 的 DER 编码（省略标签、长度和未用的位八比特组）认为是一个 **INTEGER** 的 DER 编码值。如果设置了 **BIT STRING** 的第 0 位，那么应预先计划一个额外的 0 八比特组，以便确保结果编码代表一个正的 **INTEGER**。例如：

03 02 01 f7 （代表 0-6 位集）

映射于

02 02 00 f7 （即十进制的 247）

如果 **nameSubtrees** 字段出现，那么对 **nameConstraints** 扩展中所规定的各名称形式使用相同的约定。

如果 **baseRevocationInfo** 字段出现，那么它指明 CRL 是一个关于 **PerAuthorityScope** 构件所涵盖之证书的 dCRL。使用 **crlScope** 扩展来确定 CRL 是一个 dCRL 有别于按以下方式使用 **deltaCRLIdentifier** 扩展。在 **crlScope** 情况下，**baseRevocationInfo** 部件中的信息用于指明以下时间点，即包含本扩展的 CRL 从该时间点开始提供更新。虽然是通过参考一个 CRL 来完成此任务，但所参考的 CRL 可以是也可以不是一个对适用范围而言是完整的 CRL，而 **deltaCRLIdentifier** 扩展参考的是一个已发放的 CRL，它对适用范围而言是完整的。不过，在包含 **crlScope** 扩展的 dCRL 中所提供的更新信息是对撤消信息的更新，它对适用范围而言是完整的，而不管 **baseRevocationInfo** 中所参考的 CRL 实际上是否作为一个对同一范围而言是完整的 CRL 来发放。该机制提供了比 **deltaCRLIndicator** 扩展更大的灵活性，原因是，用户可以在本地构建完整的 CRL，并基于时间进行构建，而不是基于基础 CRL（它们对适用范围而言是完整的）的发放进行构建。在这两种情况下，dCRL 都总是在一个特定的范围内、从一个特定的时间点开始为证书的撤消状态提供更新。不过，在 **deltaCRLIndicator** 情况下，该时间点是发放和参考 CRL（它对该范围而言是完整的）的时间。在 **crlScope** 情况下，该时间点是发放所参考 CRL（它对该范围而言可以是也可以不是完整的）的时间。

依赖责任机构的策略，在公布一个新的基础 CRL 之前，可以公布若干个 dCRL。包含 **crlScope** 扩展的 dCRL 在参考其构建点时，不必参考 **BaseRevocationInfo** 字段中最近发放的基础 CRL 的 **cRLNumber**。不过，在 dCRL **BaseRevocationInfo** 字段中所参考的 **cRLNumber** 应小于或等于最近发放的 CRL（对适用的范围而言是完整的）的 **cRLNumber**。

注意：**issuingDistributionPoint** 扩展和 **crlScope** 扩展相互间可能会发生冲突，未计划将它们一起使用。不过，如果 CRL 既包含 **issuingDistributionPoint** 扩展又包含 **crlScope** 扩展，那么当且仅当它满足两个扩展的准则时，公开密钥证书才落在 CRL 范围内。如果 CRL 包含 **AAissuingDistributionPoint** 扩展，但不包含 **issuingDistributionPoint** 或 **crlScope** 扩展，那么范围不包括公开密钥证书。如果 CRL 不包含 **issuingDistributionPoint**、**AAissuingDistributionPoint** 或 **crlScope** 扩展，那么范围为机构的整个范围，CRL 可用于来自该机构的任何证书。同样，**AAissuingDistributionPoint** 扩展和 **crlScope** 扩展相互间可能会发生冲突，未计划将它们一起使用。不过，如果 CRL 既包含 **AAissuingDistributionPoint** 扩展又包含 **crlScope** 扩展，那么当且仅当它满足两个扩展的准则时，属性证书才落在 CRL 范围内。如果 CRL 包含 **issuingDistributionPoint** 扩展，但不包含 **AAissuingDistributionPoint** 或 **crlScope** 扩展，那么范围不包括属性证书。如果 CRL 不包含 **issuingDistributionPoint**、**AAissuingDistributionPoint** 或 **crlScope** 扩展，那么范围为机构的整个范围，CRL 可用于来自该机构的任何证书。

当使用证书的系统使用一个包含 **crlScope** 扩展的 CRL 来检查证书状态时，它应检查感兴趣的证书和理由代码是否落在由 **crlScope** 扩展所确定的 CRL 范围内，如下所述：

a） 使用证书的系统将检查证书是否落在范围 **serialNumberRange**、**subjectKeyIdRange** 和 **nameSubtrees** 交集所指明的范围内，并与相关的 **PerAuthorityScope** 构件的 **distributionPoint** 和 **onlyContains** 相一致（如果出现的话）。

b） 如果 CRL 在 **crlScope** 扩展中包含 **onlySomeReasons** 部件，那么使用证书的系统将检查，对应用目的而言，本 CRL 所涵盖的各理由代码是否合适。如果不合适，那么可能需要额外的 CRL。注意：如果 CRL 既包含 **crlScope** 扩展又包含 **issuingDistributionPoint** 扩展，并且二者都包含 **onlySomeReasons** 部件，那么本 CRL 只涵盖包括在两个扩展 **onlySomeReasons** 部件中的那些理由代码。

8.5.2.6 状态命名扩展

本 CRL 扩展用在 CRL 结构中，作为一种向证书用户传送有关撤消通知信息的手段。这样它将出现在 CRL 结构中，因此它本身不包含任何证书撤消通知。证书用户或信赖方不得将包含本扩展的 CRL 结构作为一个撤消通知来源来使用，但可作为一种工具，用于确保使用适当的撤消信息。包含本扩展的任何 CRL 都不得作为来源，供信赖方用来检查任何证书的撤消状态。相反，信赖方可以将包含本扩展的 CRL 作为一种额外的工具来使用，用于确定适当的 CRL，以检查撤消状态。

本扩展服务于两个主要功能：

— 本扩展提供了一种用于公布信任“CRL 清单”的机制，包括所有的相关信息，以便帮助各信赖方确定，对其需求而言，它们是否拥有足够的撤消信息。例如，机构可以定期地发布新的、经过验证的 CRL 清单，典型地，拥有一个比较高的重新发布频率（相比其他 CRL 重新发布频率）。清单可以包括有关每个参考 CRL 的最新时间/日期。证书用户在得到该清单后，可以迅速地确定 CRL 的缓冲拷贝是否仍是最新的。这可以减少不必要的 CRL 检索。另外，通过使用该机制，在其通常的更新周期之间，证书用户可以知晓由机构发放的 CRL，从而改善 CRL 系统的时效性；

— 本扩展还提供了一种用于将信赖方从初始位置（例如，CRL 分发点扩展中的一个位置，或者发放机构的号码簿条目）重新指向撤消信息不同位置的机制。该特性使得机构能够对 CRL 分割方案进行修改，而不影响现有的证书或证书用户。为了做到这一点，机构将包括各个新的位置，以及将在该位置中找到的 CRL 范围。信赖方将对感兴趣的证书和范围声明进行比较，并将指针指向适当的、新的、与正在验证的证书相关的撤消信息位置。

本扩展本身是可扩展的，未来还可以使用本扩展指向其他基于非 CRL 的撤消方案。

```
statusReferrals EXTENSION ::= {
    SYNTAX              StatusReferrals
    IDENTIFIED BY       id-ce-statusReferrals }

StatusReferrals ::= SEQUENCE SIZE (1..MAX) OF StatusReferral

StatusReferral ::= CHOICE {
    cRLReferral         [0]     CRLReferral,
    otherReferral       [1]     INSTANCE OF OTHER-REFERRAL}


CRLReferral ::= SEQUENCE {
    issuer              [0]     GeneralName OPTIONAL,
    location            [1]     GeneralName OPTIONAL,
    deltaRefInfo        [2]     DeltaRefInfo OPTIONAL,
    cRLScope                    CRLScopeSyntax,
    lastUpdate          [3]     GeneralizedTime OPTIONAL,
    lastChangedCRL      [4]     GeneralizedTime OPTIONAL}
```

```
DeltaRefInfo ::= SEQUENCE {
    deltaLocation       GeneralName,
    lastDelta           GeneralizedTime OPTIONAL }

OTHER-REFERRAL ::= TYPE-IDENTIFIER
```

issuer 字段用于确定签署 CRL 的实体，它缺省为包含 CRL 的发放者名称。

location 字段提供了用于指向提名者的位置，缺省值等同于 **issuer** 名称。

deltaRefInfo 字段提供了一个可选的位置选项，从中可以获得一个 dCRL 以及一个可选的、先前 delta 的日期。

cRLScope 字段提供了 CRL 的范围，将可以在所参考的位置中找到 CRL。

lastUpdate 字段为最近发放的参考 CRL 中 **thisUpdate** 字段的值。

lastChangedCRL 字段为最近发放的参考 CRL（内容已改变）中 **thisUpdate** 字段的值。

OTHER-REFERRA 提供了扩展性，使得今后可以采用其他不是基于 CRL 的撤消方案。

本扩展总是标志为关键的，以确保使用证书的系统有意地依靠包含本扩展的 CRL，作为获得证书撤消状态信息的渠道。

如果本扩展出现，并且得到了使用证书的系统的认可，那么系统不得将 CRL 用作获得撤消状态信息的渠道。系统应该使用本扩展中所含的信息，或者使用本规范范围之外的其他方法来确定适当的撤消状态信息。

如果本扩展出现，但未得到使用证书的系统的认可，那么系统不得将 CRL 用作获得撤消状态信息的渠道。系统应该使用本规范范围之外的其他方法来确定适当的撤消信息。

8.5.2.7 CRL流标识符扩展

CRL 流标识符字段用于确认其中的 CRL 号码为惟一的正文。

```
cRLStreamIdentifier EXTENSION       ::= {
    SYNTAX              CRLStreamIdentifier
    IDENTIFIED BY       id-ce-cRLStreamIdentifier }

CRLStreamIdentifier ::= INTEGER (0..MAX)
```

本扩展总是为非关键的。

本扩展的每个值、每个机构都将是惟一的。与某个 CRL 号码相结合的 CRL 流标识符用作为惟一的标识符，用于任何指定机构发放的各个 CRL，而不管 CRL 类型是什么。

8.5.2.8 排好序的清单扩展

排好序的清单扩展用于指明 CRL 的 **revokedCertificates** 字段中的撤消证书序列是按证书序列号升序排列还是按撤消日期升序排列。本字段定义如下：

```
orderedList EXTENSION       ::= {
    SYNTAX              OrderedListSyntax
    IDENTIFIED BY       id-ce-orderedList }

OrderedListSyntax ::= ENUMERATED {
ascSerialNum        (0),
ascRevDate          (1) }
```

本扩展总是为非关键的。

- **ascSerialNum** 指明，CRL 中的撤消证书序列按证书序列号升序排列，它基于清单中每个条目 **serialNumber** 部件的值；
- **ascRevDate** 指明，CRL 中的撤消证书序列按撤消日期升序排列，它基于清单中每个条目 **revocationDate** 部件的值。

如果 **orderedList** 未出现，那么未提供任何有关排序的信息，如果有的话，用于排列 CRL 中撤消证书的清单。

8.5.2.9 Delta信息扩展

本 CRL 扩展用在不是 dCRL 的 CRL 中，向信赖方指明 dCRL 也可用于包含本扩展的 CRL。扩展提供了可以在其上找到相关 dCRL 的位置，可选地，还可提供下一个 dCRL 的发放时间。

```
deltaInfo EXTENSION       ::=     {
    SYNTAX                DeltaInformation
    IDENTIFIED BY         id-ce-deltaInfo }

DeltaInformation          ::=           SEQUENCE {
    deltaLocation         GeneralName,
    nextDelta             GeneralizedTime OPTIONAL }
```

本扩展总是为非关键的。

8.5.2.10 待撤消的扩展

本 CRL 扩展用于通知证书将在未来的某个特定日期和时间被撤消。**toBeRevoked** 扩展用于说明撤消证书的理由、撤消证书的日期和时间、将要撤消的证书组。每个清单都可以包含一个单个的证书序列号、一系列证书序列号或一个经过命名的 **subtree**。这些证书可以是公开密钥证书或者是属性证书。

```
toBeRevoked  EXTENSION  ::= {
  SYNTAX              ToBeRevokedSyntax
  IDENTIFIED BY       id-ce-toBeRevoked }

ToBeRevokedSyntax::=      SEQUENCE SIZE(1..MAX) OF ToBeRevokedGroup

ToBeRevokedGroup ::=      SEQUENCE {
  certificateIssuer   [0] GeneralName OPTIONAL,
  reasonInfo          [1] ReasonInfo       OPTIONAL,
  revocationTime          GeneralizedTime,
  certificateGroup        CertificateGroup }

ReasonInfo                ::=     SEQUENCE {
  reasonCode              CRLReason,
  holdInstructionCode     HoldInstruction OPTIONAL }

CertificateGroup     ::=  CHOICE {
  serialNumbers           [0]     CertificateSerialNumbers,
  serialNumberRange       [1]     CertificateGroupNumberRange,
  nameSubtree             [2]     GeneralName }

CertificateGroupNumberRange ::= SEQUENCE {
  startingNumber    [0]     INTEGER,
  endingNumber      [1]     INTEGER }

CertificateSerialNumbers ::= SEQUENCE SIZE(1..MAX) OF CertificateSerialNumber
```

如果出现，**certificateIssuer** 字段用于确定发放所有该 **ToBeRevokedGroup** 中所列证书的机构（CA 或 AA）。如果省略 c**ertificateIssuer**，那么它缺省为 CRL 发放者名称。

如果出现，**reasonInfo** 字段用于确定撤消证书的理由。如果出现，那么该字段指明，出于本字段中所指明的理由，将撤消在 **ToBeRevokedGroup** 中确定的所有证书。如果 **reasonCode** 包含值 **certificateHold**，那么也可以出现 **holdInstructionCode**。如果出现，那么 **holdInstructionCode** 指明当遇到 **RevokedGroup** 中所确定的任何证书时应采取什么行动。只有在 **revocationTime** 字段中所指明的撤消时间已经过去后，才可以采取这种行动。

RevocationTime 字段用于指明将在什么日期和时间撤消该组证书，并因此应被认为是无效的。该日期将比 CRL **thisUpdate** 时间要晚。如果 **revocationTime** 在包含本扩展的 CRL **thisUpdate** 时间之前，那么认为证书将在 **revocationTime** 与 **nextUpdate** 时间之间被撤消，由一个使用包含本扩展的 CRL 的信赖方来撤消。否则，它告知将在未来某个规定的时间撤消这些证书。一旦过了撤消时间，那么证书要么被 CA 撤消，要么未被撤消。如果证书被 CA 撤消了，那么今后 CRL 将在撤消证书清单中包括它，至少到证书到期为止。如果证书未被 CA 撤消，并依然想在今后撤消它，那么可以在带一个修订 **revocationTime** 的后续 CRL 的该扩展包括此证书。如果 CA 不再想撤消它，那么可以从所有的后续 CRL 中将之排除，并不得认为证书已被撤消。

certificateGroup 字段列出将要撤消的证书集。该字段用于确定由 **certificateIssuer** 中所确定之机构发放的、将要在 **revocationTime** 中所确定之日期/时间撤消的各证书。该证书集不再受任何外部控制的进一步限制（例如，**issuingDistributionPoint**）。

如果 **serialNumbers** 出现，那么带本字段中所述序列号并由确定的证书发放者发放的各证书都将在规定的时间被撤消。

如果 **serialNumberRange** 出现，那么在开始于起始序列号并终止于结束序列号范围内并由确定的证书发放者发放的所有证书都将在规定的时间被撤消。

如果 **nameSubtree** 出现，那么带对象/持有者名称（从属于规定的名称）并由确定的证书发放者发放的所有证书都将在规定的时间予以撤消。如果 **nameSubtree** 包含一个 DN，那么与公开密钥证书对象相关的所有 DNs（即 **subject** 字段和 **subjectAltNames** 扩展）或者属性证书的 **holder** 字段都将予以考虑。对其他名称形式，公开密钥证书的 **subjectAltNames** 扩展和属性证书的 **holder** 字段需加以考虑。如果证书中所含的、与对象/持有者相关的至少一个名称落在 **nameSubtree** 所规定的子树内，那么证书将规定的时间予以撤消。关于 **nameConstraints** 扩展，不是所有的名称形式对 **subtree** 规范都是合适的。在本扩展中，只有那些拥有已被认可的从属规则的名称形式才能使用。

由 CRL 发放者来决定，本扩展标志为关键的还是非关键的。由于在本扩展中提供的信息用于撤消，它将在未来发生，因此建议将之标志为非关键的，以便降低互操作性和后向兼容性问题方面的风险。

8.5.2.11 撤消的证书扩展组

可以使用以下 CRL 扩展来公布已被撤消的证书组。将要撤消的每个证书清单都关联于一个特殊的证书发放者和撤消时间。每个清单都可以包含一个证书序列号范围或一个经过命名的子树。这些证书可以是公开密钥证书，或者是属性证书。

```
revokedGroups  EXTENSION  ::= {
  SYNTAX             RevokedGroupsSyntax
  IDENTIFIED BY      id-ce-RevokedGroups }

RevokedGroupsSyntax ::=    SEQUENCE SIZE (1..MAX) OF RevokedGroup

RevokedGroup ::=    SEQUENCE {
  certificateIssuer              [0]     GeneralName OPTIONAL,
  reasonInfo                     [1]     ReasonInfo OPTIONAL,
  invalidityDate                 [2]     GeneralizedTime OPTIONAL,
  revokedcertificateGroup        [3]     RevokedCertificateGroup }

RevokedCertificateGroup ::= CHOICE {
  serialNumberRange          NumberRange,
  nameSubtree                GeneralName }
```

如果 **certificateIssuer** 字段出现，那么确定负责发放本 **RevokedGroup** 中所列之所有证书的机构（CA 或 AA）。如果省略 **certificateIssuer**，那么它缺省为 CRL 发放者名称。

如果 **reasonInfo** 字段出现，那么确定证书撤消理由。如果出现，那么本字段指明因本字段中所述之理由而撤消的、在 **RevokedGroup** 中确定的所有证书。如果 **reasonCode** 包含值 **certificateHold**，那么 **holdInstructionCode** 也可以出现。如果出现，那么 **holdInstructionCode** 指明当遇到任何在 **RevokedGroup** 中所确定的证书时应采取的行动。

如果 **invalidityDate** 字段出现，那么指明从什么时候起、在 **RevokedGroup** 中所确定的所有证书都应被认为是无效的。该日期将比 CRL **thisUpdate** 字段中所含的日期要早。如果省略，那么至少从 CRL **thisUpdate** 字段中所述的时间起，**RevokedGroup** 中所确定的所有证书都应被认为是无效的。如果在时间 **thisUpdate** 之前的证书状态对使用证书的系统而言是关键的（例如，确定在发放本 CRL 之前而证书依然有效之时或证书已经撤消之后，是否创建了一个数字签名），那么需要额外的撤消状态检查技术来确定实际的日期/时间，从该日期/时间起，某个特定证书应被认为是无效的。

revokedCertificateGroup 字段列出已被撤消的证书集。该字段用于确定由 **certificateIssuer** 中所确定之机构发放的、依据规定之条件撤消的各证书。该证书集不再受任何外部控制的进一步限制（例如，**issuingDistributionPoint**）。

如果 **serialNumberRange** 出现，那么包含规定范围内证书序列号并由确定的证书发放者发放的所有证书都

将适用。

如果 **nameSubtree** 出现，那么带对象/持有者名称（从属于规定的名称）并由确定的证书发放者发放的所有证书都将在规定的时间予以撤消。如果 **nameSubtree** 包含一个 DN，那么与公开密钥证书对象相关的所有 DNs（即 **subject** 字段和 **subjectAltNames** 扩展）或者属性证书的 **holder** 字段都将予以考虑。对其他名称形式，公开密钥证书的 **subjectAltNames** 扩展和属性证书的 **holder** 字段需加以考虑。如果证书中所含的、与对象/持有者相关的至少一个名称落在 **nameSubtree** 所规定的子树内，那么证书已被撤消。关于 **nameConstraints** 扩展，不是所有的名称形式对 subtree 规范都是合适的。在该扩展中，只有那些拥有已被认可的从属规则的名称形式才能使用。

本扩展总是标志为关键的。否则使用系统的证书可能会不正确地认为在本扩展中已确定为撤消的证书并未撤消。当本扩展出现时，它可以是 CRL 中惟一的撤消证书指示（即 **revokedCertificates** 可以为空），或者除了那些在 **revokedCertificates** 字段中指明的之外，它还可以列出已撤消的证书。一个已撤消的证书不得既列在 **revokedCertificates** 字段中，又列在本扩展中。

8.5.2.12 有关CRL扩展的过期证书

本 CRL 扩展字段指明，CRL 包括有关过期证书的撤消通知。

```
expiredCertsOnCRL EXTENSION  ::= {
  SYNTAX      ExpiredCertsOnCRL
  IDENTIFIED BY      id-ce-expiredCertsOnCRL }

ExpiredCertsOnCRL  ::= GeneralizedTime
```

本扩展总是为非关键的。

包含本扩展的 CRL 范围可以延伸，以便包括证书的撤消状态，这些证书在扩展中规定的准确时间到期，或者在该时间后到期。如果规定了对 CRL 范围的限制（通过理由代码或通过分发点），那么也适用于已经到期的证书。一旦证书已经到期，那么不得对证书的撤消状态进行更新。

8.6 CRL分发点和delta-CRL扩展

8.6.1 需求

由于撤消清单有可能变大，变得难以控制，因此需要具备表示部分 CRL 的能力。对两种不同类型的、用于处理 CRL 的执行方案，需要不同的解决方案。

第一种执行方案用在单个工作站中，它可能附带一个密码令牌。这些执行方案可能拥有有限的（如果有的话）、可信的存储容量。因此，需要对整个 CRL 进行检查，以便确定它是否有效，而后确定证书是否有效。如果 CRL 很长，那么该处理过程会很长。需要对 CRL 进行分割，以便为这些执行方案消除此问题。

第二种执行方案用在高性能服务器中，在其上处理大容量信息，例如，事务处理服务器。在该环境中，典型地，CRL 作为后台任务处理，当中，在对 CRL 进行验证后，以一种有助于加快其检查进程的表示形式，在本地保存其内容，例如，一个证书对应一位，用于指明它是否已被撤消。该表示形式保存在可信的存储器中。典型地，这种类型的服务器将需要众多机构的最新 CRL。由于它已经拥有一个先前撤消证书的清单，因此它只需检索一个新近撤消证书的清单。该清单称为 dCRL，与完整的 CRL 相比，它将更小，检索和处理所需的资源要求也将更低。

因此，以下需求与 CRL 分发点和 dCRL 有关：

a） 为了控制 CRL 的大小，需要能够指派所有证书（由一个机构发放给不同的 CRL）集的各子集。这通过将每个证书与一个 CRL 分发点相关联来实现，CRL 分发点是：

— 一个号码簿条目，其 CRL 属性将包含一个有关该证书的撤消条目，如果它已经被撤消的话；或者

— 一个位置，如电子邮件地址或国际互联网统一资源标识符（URI），从中可以获得可用的 CRL。

b） 出于性能方面的考虑，希望减少在验证多个证书时需要进行检查的 CRL 数量，例如，认证通路。这可以通过让 CRL 发放者签署并发放 CRL（包含来自多个机构的撤消）来实现。

c） 要求单独的 CRL 涵盖已撤消的机构证书和已撤消的终端实体证书。这有助于认证通路的处理，原因是，有关已撤消机构证书的 CRL 有可能非常短（通常为空）。已为此目的规定了 **authorityRevocationList** 和 **certificateRevocationList** 属性。不过，为了确保这种分隔，在 CRL 中需要有一个指示符来确定它是哪个清单。否则，不能检测到非法的、一个清单对另一个清单的替换。

d） 对潜在危害的情况（当存在巨大的专用密钥滥用风险时），需要提供一个不同的 CRL，而不是包括所有的常规绑定终点（当不存在任何大的专用密钥滥用风险时）。

e） 还需提供只包括以下证书条目的部分 CRL（即 dCRL），这些证书自基础 CRL 发放之时已被撤消。

f） 对 deltaCRL，需要指出以下日期/时间，即在此日期/时间之后该清单包含更新。

g） 要求在证书中指明，到哪里去寻找最新的 CRL（例如，最新的 delta）。

8.6.2 CRL分发点和delta-CRL扩展字段

定义了以下扩展字段：

a） *CRL 分发点*；

b） *发放分发点*；

c） *AA 发放分发点*；

d） *证书发放者*；

e） *Delta CRL 指示符*；

f） *基础更新*；

g） *最新的 CRL*。

CRL 分发点和最新的 CRL 只能当作证书扩展使用。发放分发点、AA 发放分发点、delta CRL 指示符和基础更新只能当作 CRL 扩展使用。证书发放者只能当作 CRL 条目扩展使用。

当发放分发点扩展和 AA 发放分发点扩展服务于相同目的时，它们适用于不同的证书。发放分发点扩展只适用于发放给用户与/或 CA 的公开密钥证书。AA 发放分发点扩展只适用于发放给用户和 AA 的属性证书以及发放给 SOA 的公开密钥证书。如果一个单个的 CRL 涵盖所有这些证书类型，那么该 CRL 需要包括这两种扩展。

8.6.2.1 CRL分发点扩展

CRL 分发点扩展只能当作证书扩展使用，并可以用在机构证书、终端实体公开密钥证书和属性证书中。本字段用于确定证书用户应指向哪个 CRL 分发点，以确定证书是否已被撤消。证书用户可以从适用的分发点获得一个 CRL，或者可以从机构号码簿条目获得一个当前完整的 CRL。

本字段定义如下：

```
cRLDistributionPoints EXTENSION ::= {
    SYNTAX              CRLDistPointsSyntax
    IDENTIFIED BY       id-ce-cRLDistributionPoints }

CRLDistPointsSyntax ::= 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 }

ReasonFlags ::= BIT STRING {
    unused                  (0),
    keyCompromise           (1),
    cACompromise            (2),
```

```
affiliationChanged      (3),
superseded              (4),
cessationOfOperation    (5),
certificateHold         (6),
privilegeWithdrawn      (7),
aACompromise            (8) }
```

distributionPoint 部件用于确定可以从中发现 CRL 的位置。如果本部件未出现，那么分发点名称缺省为 CRL 发放者名称。

当使用 **fullName** 可选项时，或者当使用缺省值时，分发点名称可以有多种名称形式。相同的名称，至少以其中的一种名称形式存在，将出现在 CRL 发放分发点扩展的 **distributionPoint** 字段中。不要求使用证书的系统能够处理所有的名称形式。假如能处理至少一种名称形式，那么它可以使用分发点。即使不能处理分发点的任何名称形式，假如可以从另一个渠道获得所需的撤消信息，例如，另一个分发点或机构号码簿条目，那么使用证书的系统仍可以继续使用证书。

只有当 CRL 分发点指派了一个直接从属于 CRL 发放者号码簿名称的号码簿名称时，才可以使用部件。在这种情况下，**nameRelativeToCRLIssuer** 部件用于传达有关 CRL 发放者号码簿名称的相对不同名称。

reasons 部件用于指明本 CRL 所涉及的撤消理由。如果 **reasons** 部件未出现，那么对应的 CRL 分发点分发一个 CRL，如果本证书已被撤消，那么它将包含一个有关该证书的条目。否则，**reasons** 值指明对应的 CRL 分发点涉及哪些撤消理由。

cRLIssuer 部件用于确定发放和签署 CRL 的机构。如果本部件未出现，那么 CRL 发放者名称缺省为证书发放者名称。

由证书发放者来决定，本扩展是关键的还是非关键的。出于互操作性方面的考虑，建议将之标志为非关键的。

如果本扩展标志为关键的，那么若不首先从其中一个已命名分发点（涵盖各感兴趣的理由代码）中检索和检查一个 CRL，则使用证书的系统不得使用证书。当使用分发点来为所有撤消理由代码和所有证书（由 CA 发放，包括 **cRLDistributionPoints**，作为一个关键扩展）分发 CRL 信息时，不要求 CA 还要在 CA 条目上发布一个完整的 CRL。

如果本扩展标志为非关键的，并且使用证书的系统不认可扩展字段类型，那么该系统应只使用证书，如果：

— 可以从机构获得并检查一个完整的 CRL（如果在 CRL 中未出现发放分发点扩展字段，那么表明后一个 CRL 是完整的）；

— 在本地策略下，不要求进行撤消检查；或者

— 通过其他方式来完成撤消检查。

注 1 — 可能存在由多个 CRL 发放者为一个证书发放的 CRL。协调好这些 CRL 发放者和发放机构之间的关系是一个有关机构策略方面的问题。

注 2 — 有关每个理由代码的含义在本规范第 8.5.2.2 节的理由代码字段中定义。

8.6.2.2 发放分发点扩展

本 CRL 扩展字段为该特殊 CRL 确定有关公开密钥证书的 CRL 分发点，并指明 CRL 是否是间接的，或者它是否仅限于涵盖撤消信息的一个子集。如果只使用经过分割的 CRL，那么已分割 CRL 的全集将涵盖完整的证书集，其撤消状态将利用 CRL 机制进行报告。因此，如果 CRL 发放者不使用经过分割的 CRL，那么完整的已分割 CRL 集将等同于同一证书集的一个完全 CRL。限制可以基于证书人群的子集，或者基于撤消理由的子集。CRL 通过 CRL 发放者的专用密钥进行签署 — CRL 分发点没有其自身的密钥对。不过，对一个通过号码簿进行分发的 CRL，CRL 保存在 CRL 分发点的条目中，它不可以是 CRL 发放者的号码簿条目。如果发放分发点字段、AA 发放分发点字段以及 CRL 范围字段都未出现，那么 CRL 将包含有关所有由 CRL 发放者发放的、已撤消、已到期公开密钥证书的条目。如果发放分发点字段和 CRL 范围字段都未出现，但出现了 AA 发放分发点字段，那么 CRL 的范围不包括公开密钥证书。

在证书出现在 CRL 后，在证书过期后，可以从后续的 CRL 中删除它。本字段定义如下：

```
issuingDistributionPoint EXTENSION  ::= {
    SYNTAX IssuingDistPointSyntax
    IDENTIFIED BY id-ce-issuingDistributionPoint }

IssuingDistPointSyntax ::= SEQUENCE {
    -- 如果onlyContainsUserPublicKeyCerts 和 onlyContainsCACerts 都为 FALSE,
    -- 那么CRL涵盖两种证书类型。
    distributionPoint                   [0] DistributionPointName OPTIONAL,
    onlyContainsUserPublicKeyCerts      [1] BOOLEAN DEFAULT FALSE,
    onlyContainsCACerts                 [2] BOOLEAN DEFAULT FALSE,
    onlySomeReasons                     [3] ReasonFlags OPTIONAL,
    indirectCRL                         [4] BOOLEAN DEFAULT FALSE }
```

distributionPoint 部件包含一种或多种名称形式的分发点名称。如果 **onlyContainsUserPublicKeyCerts** 为 TRUE，那么 CRL 包含有关终端实体公开密钥证书的撤消。如果 **onlyContainsCACerts** 为 TRUE，那么 CRL 包含有关 CA 证书的撤消。如果 **onlyContainsUserPublicKeyCerts** 和 **onlyContainsCACerts** 都为 FALSE，那么 CRL 包含有关终端实体公开密钥证书和 CA 证书的撤消。如果 **onlySomeReasons** 出现，那么 CRL 只包含有关已确定理由的公开密钥证书的撤消；否则，CRL 包含有关所有理由的撤消。如果 **indirectCRL** 为 TRUE，那么 CRL 可以包含有关公开密钥证书（来自机构而非来自 CRL 发放者）的撤消通知。通过该条目中证书发放者 CRL 条目扩展，或者依据第 8.6.2.3 节中所述的缺省规则，来指明负责各个条目的特殊机构。在这样一个 CRL 中，由 CRL 发放者负责确保 CRL 是完整的，这样它将包含所有的撤消条目，它们与来自所有机构（在其公开密钥证书中确定该 CRL 发放者）的 **onlyContainsUserPublicKeyCerts**、**onlyContainsCACerts** 和 **onlySomeReasons** 指示符相一致。如果通过理由代码对 CRL 进行分割，并且理由代码随撤消证书变化（导致证书从一个 CRL 流移向另一个 CRL 流），那么在所有 CRL 的 **nextUpdate** 时间之前，需要在有关旧的撤消理由的 CRL 流中继续包括证书，在有关已获得的、新的撤消理由的 CRL 流中不列出证书。

如果 CRL 包含 **issuingDistributionPoint** 扩展，并出现 **distributionPoint** 字段，那么证书中至少有一个分发点名称（例如，**cRLDistributionPoints**、**freshestCRL**、**issuer**）将匹配于 CRL 中的分发点名称。另外，它可以是以下情况，即只出现 **nameRelativeToCRLIssuer** 字段。在这种情况下，将在完全 DN（通过将 **nameRelativeToCRLIssuer** 值添加于 CRL **issuer** 字段中发现的 DN 来构建）上进行名称比较。如果正在比较的名称为 DNs（相对于 **GeneralNames** 构件中其他形式的名称），那么匹配规则用于比较两个 DNs，看它们是否相同。

对通过号码簿进行分发的 CRL，以下规则适用。如果 CRL 是一个 dCRL，那么它将通过相关分发点的 **deltaRevocationList** 属性进行分发，或者如果没有确定任何分发点，那么通过 CRL 发放者条目的 **deltaRevocationList** 属性进行分发，而不管 CRL 所涵盖的证书类型的设置是什么。除非 CRL 是一个 dCRL：

- 设置了 **onlyContainsCACerts** 并且不包含 **AAissuingDistributionPoint** 扩展的 CRL，将通过相关分发点的 **authorityRevocationList** 属性进行分发，或者如果未确定任何分发点，那么通过 CRL 发放者条目的 **authorityRevocationList** 属性进行分发；
- 设置了 **onlyContainsCACerts** 并且包含 **AAissuingDistributionPoint** 扩展（**containsUserAttributeCerts** 设为 FALSE）的 CRL，将通过相关分发点的 **authorityRevocationList** 属性进行分发，或者如果未确定任何分发点，那么通过 CRL 发放者条目的 **authorityRevocationList** 属性进行分发；
- 只将 **onlyContainsCACerts** 设为了 FALSE 的 CRL，将通过相关分发点的 **certificateRevocationList** 属性进行分发，或者如果未确定任何分发点，那么通过 CRL 发放者条目的 **certificateRevocationList** 属性进行分发；
- 既包含 **issuingDistributionPoint** 扩展，又包含 **AAissuingDistributionPoint** 扩展（带 **containsUserAttributeCerts** 设置）的 CRL，将通过相关分发点的 **certificateRevocationList** 属性进行分发，或者如果未确定任何分发点，那么通过 CRL 发放者条目的 **certificateRevocationList** 属性进行分发。

本扩展总是为关键的。不理解本扩展的证书用户不能假定 CRL 包含一个完整的、已确定机构的撤消证书清单。不包含关键扩展的 CRL 将包含所有当前的、有关发放机构的 CRL 条目，包括有关所有已撤消用户证书和机构证书的条目。

注 1 — 机构向 CRL 发放者传送撤消信息的方式超出了本号码簿规范的讨论范围。

注 2 — 如果机构公布了一个 CRL，并将 **onlyContainsUserPublicKeyCerts** 或 **onlyContainsCACerts** 设为了 true，那么机构应确保该 CRL 覆盖的所有 CA 证书都包含 **basicConstraints** 扩展。

8.6.2.3 证书发放者扩展

本 CRL 条目扩展用于确定与间接 CRL 中某个条目相关的证书发放者，即在其发放分发点扩展中设置了 **indirectCRL** 指示符。如果本扩展未出现在间接 CRL 的第一个条目中，那么证书发放者缺省为 CRL 发放者。在间接 CRL 的后续条目上，如果本扩展未出现，那么条目的证书发放者等同于先前条目的证书发放者。

本字段定义如下：

```
certificateIssuer EXTENSION ::= {
    SYNTAX              GeneralNames
    IDENTIFIED BY       id-ce-certificateIssuer }
```

本扩展总是为关键的。如果某个实施忽略了本扩展，那么它可能不能正确地将 CRL 条目施加于证书上。

8.6.2.4 Delta CRL指示符扩展

Delta CRL 指示符字段用于确定 CRL 是一个 delta CRL（dCRL），它用于更新所参考的基础 CRL。所参考的基础 CRL 是这样一个 CRL，即它将明确地当作一个对某个特定范围而言是完整的 CRL 来发放。包含 delta CRL 指示符扩展的 CRL 包含对同一范围内证书撤消状态所做的更新。该范围不必包括所有的撤消理由，或者所有的、由 CA 发放的证书，尤其在以下情况下，即 CRL 是一个 CRL 分发点。不过，对适用的范围而言，在发布 dCRL 之时，包含 delta CRL 指示符扩展的 CRL 与本扩展 **BaseCRLNumber** 部件中所参考的 CRL 相结合将等同于一个完整的 CRL。

本字段定义如下：

```
deltaCRLIndicator EXTENSION ::= {
    SYNTAX              BaseCRLNumber
    IDENTIFIED BY       id-ce-deltaCRLIndicator }

BaseCRLNumber ::= CRLNumber
```

BaseCRLNumber 类型的值用于确定基础 CRL 的 CRL 号码，基础 CRL 用作产生该 dCRL 的基础。所参考的 CRL 将是一个对适用范围而言是完整的 CRL。

本扩展总是为关键的。不理解 dCRL 用途的证书用户不得使用包含本扩展的 CRL，原因是，CRL 可能并不像用户所期望的那样是完整的。

8.6.2.5 基础更新扩展

基础更新字段用在 dCRL 中，用于确定以下日期/时间，即在该日期/时间后，本 delta 对撤消状态进行更新。本扩展只应用在包含 **deltaCRLIndicator** 扩展的 dCRL 中。一个包含 **crlScope** 扩展的 dCRL 不需要本扩展，原因是，**crlScope** 扩展的 **baseThisUpdate** 字段可用于同一目的。

```
baseUpdateTime EXTENSION        ::= {
    SYNTAX              GeneralizedTime
    IDENTIFIED BY       id-ce-baseUpdateTime }
```

本扩展总是为非关键的。

8.6.2.6 最新的CRL扩展

最新的 CRL 扩展可以用作一个证书或 CRL 扩展。在证书内，本扩展可以在发放给机构的证书中或者发放给用户的证书中使用。本字段用于确定以下 CRL，即证书用户将参考它来获得最新的撤消信息（例如，最新的 dCRL）。本字段定义如下：

```
freshestCRL EXTENSION     ::= {
    SYNTAX              CRLDistPointsSyntax
    IDENTIFIED BY       id-ce-freshestCRL }
```

类型 **CRLDistPointsSyntax** 的值在第 8.6.2.1 节的 CRL 分发点扩展中进行定义。

由证书发放者来决定，本扩展是关键的还是非关键的。如果最新的 CRL 扩展被标志为关键的，那么若不首先检索和检查最新的 CRL，则证书使用系统不得使用证书。如果扩展被标志为非关键的，那么证书使用系统可

以使用本地方法来确定是否要求对最新的 CRL 进行检查。

8.6.2.7 AA发放分发点扩展

本 CRL 扩展字段用于确定有关本特殊 CRL 的属性证书的 CRL 分发点，并指明 CRL 是否是间接的，或者它只限于涵盖撤消信息的某个子集。限制可以基于证书人群的子集，或者基于撤消理由的子集。CRL 通过 CRL 发放者的专用密钥进行签署—CRL 分发点没有其自身的密钥对。不过，对一个通过号码簿进行分发的 CRL，CRL 保存在 CRL 分发点的条目中，它不可以是 CRL 发放者的号码簿条目。如果发放分发点扩展、AA 发放分发点扩展以及 CRL 范围字段都未出现，那么 CRL 将包含有关所有由 CRL 发放者发放的、已撤消、已到期属性证书的条目。如果 AA 发放分发点字段和 CRL 范围字段都未出现，但出现了发放分发点字段，那么 CRL 的范围不包括属性证书。

在证书出现在 CRL 后，在证书过期后，可以从后续的 CRL 中删除它。

本字段定义如下：

```
AAissuingDistributionPoint EXTENSION  ::= {
    SYNTAX AAIssuingDistPointSyntax
    IDENTIFIED BY id-ce-AAissuingDistributionPoint }

AAIssuingDistPointSyntax  ::= SEQUENCE {
    distributionPoint           [ 0 ] DistributionPointName OPTIONAL,
    onlySomeReasons             [ 1 ] ReasonFlags OPTIONAL,
    indirectCRL                 [ 2 ] BOOLEAN DEFAULT FALSE,
    containsUserAttributeCerts  [ 3 ] BOOLEAN DEFAULT TRUE,
    containsAACerts             [ 4 ] BOOLEAN DEFAULT TRUE,
    containsSOAPublicKeyCerts   [ 5 ] BOOLEAN DEFAULT TRUE }
```

distributionPoint 部件包含一种或多种名称形式的分发点名称。如果 **onlySomeReasons** 出现，那么 CRL 只包含有关已确定理由的属性证书的撤消；否则，CRL 包含有关所有理由的撤消。

如果 **indirectCRL** 为 TRUE，那么 CRL 可以包含有关属性证书（来自机构而非来自 CRL 发放者）的撤消通知。通过该条目中证书发放者 CRL 条目扩展，或者依据第 8.6.2.3 节中所述的缺省规则，来指明负责各个条目的特殊机构。在这样一个 CRL 中，由 CRL 发放者负责确保 CRL 是完整的，这样它将包含所有的撤消条目，它们与来自所有机构（在其属性证书中确定该 CRL 发放者）的 **containsUserAttributeCerts**、**containsAACerts**、**containsSOAPublicKeyCerts** 和 **onlySomeReasons** 指示符相一致。

如果 **containsUserAttributeCerts** 为 TRUE，那么 CRL 包含有关属性证书（发放给本身不是 AA 的终端实体）的撤消。如果 **containsAACerts** 为 TRUE，那么 CRL 包含有关属性证书（发放给本身是 AA 的对象）的撤消。

如果 **containsSOAPublicKeyCerts** 为 TRUE，那么 CRL 包含有关公开密钥证书（出于特权管理目的，发放给是 SOA 的实体，即证书包含 **SOAIdentifier** 扩展）的撤消。对通过号码簿进行分发的 CRL，以下规则适用。如果 CRL 是一个 dCRL，那么它将通过相关分发点的 **deltaRevocationList** 属性进行分发，或者如果没有确定任何分发点，那么通过 CRL 发放者条目的 **deltaRevocationList** 属性进行分发，而不管 CRL 所涵盖的证书类型的设置是什么。除非 CRL 是一个 dCRL：

- 不包含 **issuingDistributionPoint** 扩展（它只有 **containsAACerts** 与/或 **containsSOAPublicKeyCerts** 集）的 CRL，将通过相关分发点的 **attributeAuthorityRevocationList** 属性进行分发，或者如果没有确定任何分发点，那么通过 CRL 发放者条目的 **attributeAuthorityRevocationList** 属性进行分发；
- 不包含 **issuingDistributionPoint** 扩展（它有 **containsUserAttributeCerts** 集，有或没有 **containsAACerts** 与/或 **containsSOAPublicKeyCerts** 集）的 CRL，将通过相关分发点的 **attributeCertificateRevocationList** 属性进行分发，或者如果没有确定任何分发点，那么通过 CRL 发放者条目的 **attributeCertificateRevocationList** 属性进行分发；
- 包含 **issuingDistributionPoint** 扩展的 CRL，将按第 8.6.2.2 节的规定进行分发。

本扩展总是为关键的。不理解本扩展的证书用户不能假设 CRL 包含一个完整的、有关已确定机构撤消证书的清单。不包含关键扩展的 CRL 将包含有关发放机构的所有当前 CRL 条目，包括有关所有已撤消用户证书和机构证书的条目。

注 1 — 机构向 CRL 发放者传送撤消信息的方式超出了本号码簿规范的讨论范围。

注 2 — 如果机构公布了一个 CRL，并将 **containsAACerts** 设为了 **true**，未将 **containsUserAttributeCerts** 设为 **true**，那么机构应确保该 CRL 覆盖的所有 AA 证书都包含 **basicAttConstraints** 扩展。

注 3 — 如果机构公布了一个 CRL，并将 **containsSOAPublicKeyCerts** 设为了 **true**，那么机构应确保该 CRL 覆盖的所有 SOA 证书都包含 **SOAIdentifier** 扩展。

9 Delta CRL与基础CRL之间的关系

一个 dCRL 包括一个 **deltaCRLIndicator** 或者一个 **crlScope** 扩展，用于指明在本 dCRL 中进行更新的基础撤消信息。

如果 **deltaCRLIndicator** 出现在一个 dCRL 中，那么正在更新的基础撤消信息为在该扩展中参考的基础 CRL。**deltaCRLIndicator** 扩展所参考的基础 CRL 将是这样一个 CRL，即它对其范围是完整的（即它本身不是一个 dCRL）。

如果 **crlScope** 扩展出现，并包含 **baseRevocationInfo** 部件，以参考正在更新的基础撤消信息，那么从该 dCRL 提供更新之时起，它便是一个对某个特殊点的参考。**baseRevocationInfo** 部件参考了一个 CRL，对该范围它可以是也可以不是完整的（即所参考的 CRL 可以只作为一个 dCRL 发放）。不过，dCRL（包含 **baseRevocationInfo** 部件）对撤消信息进行更新，在发放所参考的 CRL 之时，对所参考之 CRL 的范围而言，它是完整的。证书用户可以将 dCRL 用于一个对给定范围而言是完整的 CRL，它在同一时间发放，或者在发放了 dCRL（包含 **baseRevocationInfo** 部件）中所参考的 CRL 之后才发放。

由于可能与信息产生冲突，因此一个 CRL 不得既包含 **deltaCRLIndicator** 扩展，又包含带 **baseRevocationInfo** 部件的 **crlScope** 扩展。只有当 **baseRevocationInfo** 部件不出现在 **crlScope** 扩展中时，一个 CRL 才可以既包含 **deltaCRLIndicator** 扩展，又包含 **crlScope** 扩展，

一个 dCRL 也可以是一个间接的 CRL，这种情况下，它可以包含经过更新的、与一个或多个机构发放的基础 CRL 相关的撤消信息。**crlScope** 扩展将用做一种方法来确定一个 CRL 是一个间接的 dCRL。**crlScope** 扩展将为每个基础 CRL 包含一个 **PerAuthorityScope** 部件的实例，间接的 dCRL 为每个基础 CRL 提供更新信息。

将 dCRL 应用于所参考的基础撤消信息将准确反映当前的撤消状态。

— 一个带撤消理由 **certificateHold** 的证书撤消通知可以出现在一个 dCRL 中，或出现在一个对给定范围而言是完整的 CRL 中。在尚未进一步决定是永久撤消证书还是将其恢复为未撤消状态之时，该理由代码用于指明临时撤消证书。

  - 如果在 CRL（为一个对给定范围而言是完整的 dCRL 或 CRL）中证书显示为已被撤消，撤消理由为 **certificateHold**，其 **cRLNumber** 为 *n*，随后放弃持有，那么证书将包括在所有在释放持有之后发放的 dCRL 中，当中，所参考的基础 CRL 的 **cRLNumber** 小于或等于 *n*。依赖用于指明该 CRL 是一个 dCRL 的扩展，所参考的基础 CRL 的 CRL 数量为 **deltaCRLIndicator** 扩展 **BaseCRLNumber** 部件的值，或者为 **cRLScope** 扩展 **BaseRevocationInfo** 部件的 **cRLNumber** 元素。证书将显示撤消理由 **removeFromCRL**，除非在此之后，出于 dCRL 所涵盖的撤消理由之一，再次撤消证书，在这种情况下，证书将显示有关后续撤消的撤消理由。
  - 如果证书未被释放持有，但被永久地撤消，那么它将列于所有的后续 dCRL 中，当中，所参考的基础 CRL 的 **cRLNumber** 小于 CRL（为一个对给定范围而言是完整的 dCRL 或 CRL）的 **cRLNumber**，永久撤消通知首先出现在其上。依赖用于指明该 CRL 是一个 dCRL 的扩展，所参考的基础 CRL 的 CRL 数量为 **deltaCRLIndicator** 扩展 **BaseCRLNumber** 部件的值，或者为 **cRLScope** 扩展 **BaseRevocationInfo** 部件的 **cRLNumber** 元素。

— 一个证书撤消通知可以首先出现在 dCRL 中，有可能在发放下一个对适用范围而言是完成的 CRL 之前，证书的有效期已经到期。在这种情况下，该撤消通知将包括在所有后续的 dCRL 中，直至该撤消通知包括在至少一个已经发放的、对该证书范围而言是完整的 CRL 中。

对某个特定范围完整的 CRL，当前可以在本地用以下方法之一来构造：

— 通过在该范围内检索当前的 dCRL，并将之与一个发放的 CRL 相结合，此 CRL 对该范围是完整的，其 **cRLNumber** 大于或等于 dCRL 中所参考的基础 CRL 的 **cRLNumber**；或者

— 通过在该范围内检索当前的 dCRL，并将之与一个本地构建的 CRL 相结合，此 CRL 对该范围是完整的，它用一个 dCRL 来构建，此 dCRL 的 **cRLNumber** 大于或等于当前 dCRL 中所参考的基础 CRL 的 **cRLNumber**。

10 认证通路处理程序

认证通路处理在需要使用远程终端实体公开密钥的系统中完成，例如一个对远程实体产生的数字签名进行验证的系统。设计证书策略、基本约束、名称约束和策略约束扩展是为了推动认证通路处理逻辑的自动、自包含执行。

以下是对认证通路确认处理程序的一个概述。功能上，执行方案应等同于由此程序产生的外部行为。某个特殊执行方案用于从给定输入获得正确输出的算法不是标准化的。

10.1 通路处理输入

认证通路处理程序的输入为：

a） 由一系列证书组成的一条认证通路；

注 — 认证通路中的每个证书都是惟一的。一条两次或更多次包含相同证书的通路不是一条有效的认证通路。

b） 信任的公开密钥或密钥标识符（如果密钥在内部保存于认证通路处理模块中），用于验证认证通路中的第一个证书；

c） 由一个或多个证书策略标识符组成的 *initial-policy-set*，指明出于认证通路处理的目的，这些策略中的任何一个都将适用于证书用户；该输入还可以取特殊值 *any-policy*，但它不能为空；

d） 一个 *initial-explicit-policy* 指示符值，它指明一个可接受的策略标识符是否需要显性地出现在通路中所有证书的证书策略扩展字段中；

e） 一个 *initial-policy-mapping-inhibit* 指示符值，它指明在认证通路中是否禁止策略映射；

f） 一个 *initial-inhibit-policy* 指示符值，它指明如果出现在证书策略扩展中，是否认为特殊值 **anyPolicy** 匹配于约束集中的任何特殊证书策略；

g） 当前的日期/时间（如果在内部不可供认证通路处理模块使用）；

h） 一个 *initial-permitted-subtrees-set*，它包含一个用于定义子树的子树规范集，在其中允许对象名称（名称形式为用于规定子树的形式）。在认证通路的各证书中，具有某种特定名称形式的所有对象名称（对其定义了最初的许可子树）将落在具有该特定名称形式的许可子树集中。该输入还可以包含没有限制的特殊值，用于指明所有的对象名称在一开始都是可接受的。对第 10 节，对象名称为那些出现在对象字段或 *subjectAltName* 扩展中的名称值；

i） 一个 *initial-excluded-subtrees-set*，它包含一个用于定义子树的初始子树规范集，认证通路证书中的对象名称不得落在其中。该输入还可以是一个空集，指明一开始任何子树排斥行为都不起作用；

j） 一个 *initial-required-name-forms*，它包含一个最初的名称形式集，指明通路中的所有证书都必须包括一个对象名称，形式至少为一种规定的名称形式。该输入还可以是一个空集，指明对证书中的对象名称没有任何特殊的名称形式要求。

c）、d）、e）和 f）的值取决于用户应用组合的策略需求，需要使用经过认证的终端实体公开密钥。

注意：由于这些是通路确认处理的各单独输入，因此证书用户可以将其置于任何特定信任公开密钥上的信任限制于一系列特定的证书策略上。这可以通过以下方式实现，即确保只有当 *initial-policy-set* 输入包括证书用户信任该公开密钥的策略时，某个特定的公开密钥才能作为处理过程的输入。由于处理过程的另一个输入为认证通路

本身，因此可以基于事务处理，在事务处理中实施该控制。

10.2 通路处理输出

程序的输出为：

a） 一个指示，指出认证通路确认是成功还是失败；

b） 如果确认失败，那么是一个诊断代码，指明失败的理由；

c） 机构限制的策略集及其相关的限定符，依据有效的认证通路，或特殊值 *any-policy*；

d） 用户限制的策略集，来自 *authorities-constrained-policy-set* 和 *initial-policy-set* 的交集；

e） *explicit-policy-indicator*，指明是通路中的认证用户还是机构要求在通路中的每个证书中确定可接受的策略；以及

f） 发生在认证通路处理中的任何策略映射的细节。

注 — 如果验证成功，那么作为策略限定符值或证书中其他信息的结果，使用证书的系统可以继续选择不使用证书，

10.3 通路处理变量

程序使用以下状态变量集：

a） *authorities-constrained-policy-set*：来自认证通路证书的策略标识符和限定符表（行代表策略、其限定符和映射历史，列代表认证通路中的证书）；

b） *permitted-subtrees*：用于定义子树的子树规范集，认证通路各后续证书中的所有对象名称都需要落在其中，或者可以取特殊值 *unbounded*；

c） *excluded-subtrees*：用于定义子树的子树规范集（可能为空）（每个包含一个子树基础名称以及最高级和最低级指示符），认证通路后续证书中的任何对象名称都不可以落在其中；

d） *required-name-forms*：名称形式集中的一个集（可能为空）。对每个名称形式集，每个后续证书都必须包含一个具有集中其中一种名称形式的名称；

e） *explicit-policy-indicator*：指明是否需要在通路的每个证书中都显性确定一个可接受的策略；

f） *path depth*：一个整数，比已完成处理的认证通路中的证书数量多 1；

g） *policy-mapping-inhibit-indicator*：指明是否禁止策略映射；

h） *inhibit-any-policy-indicator*：指明是否认为特殊值 **anyPolicy** 匹配于任何特殊的证书策略；

i） *pending-constraints*：有关 *explicit-policy* *inhibit-policy-mapping* 与/或 *inhibit-any-policy* 约束的细节，已对它们做了规定，但尚未起作用。有三个 1 位指示符，称为 *explicit-policy-pending*、*policy-mapping-inhibit-pending* 和 *inhibit-any-policy-pending*，对每位，有一个称为 *skip-certificates* 的整数，给出了需要在约束起作用之前跳过的证书数量。

10.4 初始化步骤

程序涉及一个初始化步骤，后跟一系列证书处理步骤。初始化步骤包括：

a） 在 *authorities-constrained-policy-set* 表的第 0 行第 0 列和第 1 列写入 *any-policy*；

b） 将 *permitted-subtrees* 变量初始化为 *initial-permitted-subtrees-set* 值；

c） 将 *excluded-subtrees* 变量初始化为 *initial-excluded-subtrees-set* 值；

d） 将 *required-name-forms* 变量初始化为 *initial-required-name-forms* 值；

e） 将 *explicit-policy-indicato* 初始化为 *initial-explicit-policy* 值；

f） 将 *path-depth* 初始化为 1；

g） 将 *policy-mapping-inhibit-indicator* 初始化为 *initial-policy-mapping-inhibit* 值；

h） 将 *inhibit-any-policy-indicato* 初始化为 *initial-inhibit-policy* 值；

i） 将三个 *pending-constraints* 指示符初始化为未设置。

10.5 证书处理

而后依次对每个证书进行处理，从利用输入信任的公开密钥进行签署的证书开始。最后一个证书被认为是最终的证书；任何其他证书都被认为是中间证书。

10.5.1 基本的证书检查

对证书进行以下检查。如果在通路中遇到自发放证书，那么忽略之。

a） 检查签名是否得到了验证，日期是否有效，证书对象和证书发放者名称是否得到了正确链接，证书是否未被撤消。

b） 对一个中间的第 3 版本证书，检查 **basicConstraints** 是否出现，扩展 **basicConstraints** 中的 **cA** 部件是否为 **TRUE**。如果 **pathLenConstraint** 部件出现，那么检查当前的认证通路是否不与该约束存在冲突（忽略中间的自发放证书）。

c） 如果证书策略扩展未出现，那么通过删去 *authorities-constrained-policy-set* 表中的所有行，来将 *authorities-constrained-policy-set* 设为空。

d） 如果证书策略扩展出现，那么对扩展而非 **anyPolicy** 中的每个策略 *P*，那么将与 *P* 相关的策略限定符绑定于 *authorities-constrained-policy-set* 表中的每一行上，表的[*path-depth*]列条目包含值 *P*。如果 *authorities-constrained-policy-set* 表中的任何一行都不包含其[*path-depth*]列条目中的 *P*，但 *authorities-constrained-policy-set*[0, *path-depth*]中的值为 *any-policy*，那么通过复制第 0 行，为表增加一个新的行，并写入策略标识符 *P* 及其新行[*path-depth*]列条目中的限定符。

e） 如果证书策略扩展出现，并且不包括值 **anyPolicy**，或者如果设置了 *inhibit-any-policy-indicator*，并且证书不是一个自发放的中间证书，那么将删去其[*path-depth*]列条目包含值 *any-policy* 的任何行，以及其[*path-depth*]列条目不包含证书策略扩展中某个值的任何行。

f） 如果证书策略扩展出现，并且包括值 **anyPolicy**，以及未设置 *inhibit-any-policy-indicator*，那么将与 **anyPolicy** 相关的策略限定符绑定于 *authorities-constrained-policy-set* 表中的每一行上，表的[*path-depth*]列条目包含值 *any-policy*，或者包含一个未出现在证书策略扩展中的值。

g） 如果证书不是一个中间的自发放证书，那么检查对象名称是否在由 *permitted-subtrees* 值给出的名称空间中，并且不在由 *excluded-subtrees* 值给出的名称空间中。

h） 如果证书不是一个中间的自发放证书，并且如果 *required-name-forms* 不是一个空集，那么对 *required-name-forms* 中的每个名称形式集，检查在集的证书（具有其中一种名称形式）中是否存在一个对象名称。

10.5.2 处理中间证书

对中间证书，而后执行以下约束记录行动，以便为下一个证书的处理正确设置状态变量。如果在通路中遇到自发放证书，那么忽略之。

a） 如果带 **permittedSubtrees** 部件的 **nameConstraints** 扩展出现在证书中，那么将 *permitted-subtrees* 状态变量设为其先前值与证书扩展中指明值的交集。

b） 如果带 **excludedSubtrees** 部件的 **nameConstraints** 扩展出现在证书中，那么将 *excluded-subtrees* 状态变量设为其先前值与证书扩展中指明值的并集。

c） 如果带 **requiredNameForms** 部件的 **nameConstraints** 扩展出现在证书中，那么将 required-name-forms 变量设为其先前值与由证书扩展中规定的名称形式集组成的集的并集。如果 **requiredNameForms** 部件包含多个名称形式，那么 required-name-forms 变量将指出，在所有的后续证书中应出现一个名称，它具有至少一种本扩展中所指的名称形式。required-name-forms 变量值

与当前证书扩展值的并集是一个集，用于为所有的后续证书设置信令需求。例如，如果设置当前的 required-name-forms，要求一个 DN 或者一个 rfc822 名称必须出现在证书中，并且正在处理的证书中的当前扩展指明，需要 rfc822 名称或 DNS 名称，那么作为新的 required-name-forms 的结果并集将指明，每个后续证书都必须有一个 rfc822 名称，或者既有一个 DN，又有一个 DNS 名称。

d） 如果设置了 *policy-mapping-inhibit-indicator*：

— 对任何策略映射扩展进行处理，对扩展中确定的每个映射，确定 *authorities-constrained-policy-set* 表中的所有行，其[*path-depth*]列条目等于扩展中的发放者域策略值，并删除行。

e） 如果未设置 *policy-mapping-inhibit-indicator*：

— 对任何策略映射扩展进行处理，对扩展中确定的每个映射，确定 *authorities-constrained-policy-set* 表中的所有行，其[*path-depth*]列条目等于扩展中的发放者域策略值，并将同一行[*path-depth*+1]列条目中的扩展写入对象域策略值中。如果扩展将一个发放者域策略映射于多个对象域策略，那么拷贝受影响的行，并将新的条目加入每个行。如果 *authorities-constrained-policy-set*[0, *path-depth*]中的值为 *any-policy*，那么将[*path-depth*]列中的策略映射扩展写入每个发放者域策略标识符中，必要的话，对行进行复制，如果它们出现，那么保存标识符，并将同一行[*path-depth*+1]列条目中的扩展写入对象域策略值中。

— 如果设置了 *policy-mapping-inhibit-pending* 指示符，并且证书不是自发放的，那么递减对应的 *skip-certificates* 值，如果该值变为 0，那么设置 *policy-mapping-inhibit-indicator*。

— 如果 **inhibitPolicyMapping** 约束出现在证书中，那么执行以下内容。对为 0 的 **SkipCerts** 值，设置 *policy-mapping-inhibit-indicator*。对任何其他的 **SkipCerts** 值，设置 *policy-mapping-inhibit-pending* 指示符，并将对应的 *skip-certificates* 值设为 **SkipCerts** 值和先前 *skip-certificates* 值中的较小者（如果已经设置了 *policy-mapping-inhibit-pending* 指示符的话）。

f） 对在上面步骤 c）或 d）中未做修改的任何行（以及对在证书中未出现任何映射扩展情况下的每一行），将该行[*path-depth*]列中的策略标识符写入[*path-depth*+1]列中。

g） 如果未设置 *inhibit-any-policy-indicator*：

— 如果设置了 *inhibit-any-policy-pending* 指示符，并且证书不是自发放的，那么递减对应的 *skip-certificates* 值，如果该值变为 0，那么设置 *inhibit-any-policy-indicator*。

— 如果 **inhibitAnyPolicy** 约束出现在证书中，那么执行以下内容。对为 0 的 **SkipCerts** 值，设置 *inhibit-any-policy-indicator*。对任何其他的 **SkipCerts** 值，设置 *inhibit-any-policy-pending* 指示符，并将对应的 *skip-certificates* 值设为 **SkipCerts** 值和先前 *skip-certificates* 值中的较小者（如果已经设置了 *inhibit-any-policy-pending* 指示符的话）。

h） 递增[*path-depth*]。

10.5.3 显性策略指示符处理

对所有证书，之后执行以下操作：

a） 如果未设置 *explicit-policy-indicator*：

— 如果设置了 *explicit-policy-pending* 指示符，并且证书不是自发放的中间证书，那么递减对应的 *skip-certificates* 值，如果该值变为 0，那么设置 *explicit-policy-indicator*。

— 如果 **requireExplicitPolicy** 约束出现在证书中，那么执行以下内容。对为 0 的 **SkipCerts** 值，设置 *explicit-policy-indicator*。对任何其他的 **SkipCerts** 值，设置 *explicit-policy-pending* 指示符，并将对应的 *skip-certificates* 值设为 **SkipCerts** 值和先前 *skip-certificates* 值中的较小者（如果已经设置了 *explicit-policy-pending* 指示符的话）。

— 如果 **requireExplicitPolicy** 部件出现，并且认证通路包括一个由已命名 CA 发放的证书，那么对通路中的所有证书，都需要在证书策略扩展中包含一个可接受的策略标识符。一个可接受的

策略标识符是认证通路用户要求的证书策略标识符，通过策略映射或 *any-policy*，已宣布策略标识符等同于它。已命名 CA 是包含本扩展的证书的发放者 CA（如果 **requireExplicitPolicy** 的值为 0 的话），或者是一个 CA，它是认证通路中后续证书的对象（通过一个非 0 的值来指明）。

10.5.4 最后的处理

一旦通路中的所有证书已经经过处理，那么之后执行以下操作：

a） 从 *authorities-constrained-policy-set* 表来确定 *authorities-constrained-policy-set*。如果表为空，那么 *authorities-constrained-policy-set* 为空或未设置。如果 *authorities-constrained-policy-set[0, path-depth]* 为 *any-policy*，那么 *authorities-constrained-policy-set* 为 *any-policy*。否则，对表中的每一行，*authorities-constrained-policy-set* 为最左边单元中的值，它不包含标识符 *any-policy*。

b） 通过形成 *authorities-constrained-policy-set* 和 *initial-policy-set* 的交集，计算 *user-constrained-policy-set*。

c） 如果设置了 *explicit-policy-indicator*，那么检查 *authorities-constrained-policy-set* 和 *user-constrained-policy-set* 是否都不为空。

如果任何上述检查都失败，那么程序将终止，返回一个失败指示、一个适当的理由代码、*explicit-policy-indicator*、*authorities-constrained-policy-set* 和 *user-constrained-policy-set*。如果是由于一个空的 *user-constrained-policy-set* 而造成检查失败，那么在机构约束的策略条件下，通路是有效的，但对用户而言，没有一个是可接受的。

如果对最终证书，任何上述检查都未失败，那么程序程序将终止，返回一个成功指示，以及 *explicit-policy-indicator*、*authorities-constrained-policy-set* 和 *user-constrained-policy-set*。

11 PKI号码簿方案

本节定义了号码簿方案元素，用于代表号码簿中的 PKI 信息。它包括相关对象类别、属性和属性值匹配规则的规范。

11.1 PKI号码簿对象类别和名称形式

本子节包括对象类别的定义，用于代表号码簿中的 PKI 对象。

11.1.1 PKI用户对象类别

PKI 用户对象类别用于定义有关对象的条目，它们可以是公开密钥证书的对象。

```
pkiUser     OBJECT-CLASS ::= {
    SUBCLASS OF         {top}
    KIND                auxiliary
    MAY CONTAIN         {userCertificate}
    ID                  id-oc-pkiUser }
```

11.1.2 PKI CA对象类别

PKI CA 对象类别用于定义有关作为认证机构的对象的条目。

```
pkiCA    OBJECT-CLASS          ::= {
         SUBCLASS OF       {top}
         KIND              auxiliary
         MAY CONTAIN       {cACertificate |
                           certificateRevocationList |
                           authorityRevocationList |
                           crossCertificatePair }
         ID                id-oc-pkiCA }
```

11.1.3 CRL分发点对象类别和名称形式

CRL 分发点对象类别用于定义有关作为 CRL 分发点的对象的条目。

```
cRLDistributionPoint     OBJECT-CLASS          ::= {
    SUBCLASS OF          { top }
    KIND                 structural
    MUST CONTAIN         { commonName }
```

```
    MAY CONTAIN         { certificateRevocationList |
                          authorityRevocationList |
                          deltaRevocationList }
    ID                  id-oc-cRLDistributionPoint }
```

CRL 分发点名称形式规定了如何命名对象类别 **cRLDistributionPoint** 的条目。

```
cRLDistPtNameForm       NAME-FORM     ::= {
    NAMES               cRLDistributionPoint
    WITH ATTRIBUTES     { commonName}
    ID                  id-nf-cRLDistPtNameForm }
```

11.1.4 Delta CRL对象类别

delta CRL 对象类别用于定义有关持有 delta 撤消清单（如 CA、AA 等）的对象的条目。

```
deltaCRL                OBJECT-CLASS ::= {
    SUBCLASS OF         {top}
    KIND                auxiliary
    MAY CONTAIN         {deltaRevocationList}
    ID                  id-oc-deltaCRL }
```

11.1.5 证书策略和CPS对象类别

CP CPS 对象类别用于定义有关包含证书策略与/或认证实施信息的对象的条目。

```
cpCps                   OBJECT-CLASS ::= {
    SUBCLASS OF         {top}
    KIND                auxiliary
    MAY CONTAIN         {certificatePolicy |
                         certificationPracticeStmt}
     ID                 id-oc-cpCps }
```

11.1.6 PKI认证通路对象类别

PKI 认证通路对象类别用于定义有关包含 PKI 通路的对象的条目。通常它将用于连接结构化的 **pkiCA** 或 **pkiUser** 的条目。

```
pkiCertPath             OBJECT-CLASS ::= {
    SUBCLASS OF         {top}
    KIND                auxiliary
    MAY CONTAIN         { pkiPath }
    ID                  id-oc-pkiCertPath }
```

11.2 PKI号码簿属性

本子节包括号码簿属性的定义，用于保存号码簿中的 PKI 信息元素。

11.2.1 用户证书属性

一个用户可以从一个或多个 CA 获得一个或多个公开密钥证书。**userCertificate** 属性类型包含用户已从一个或多个 CA 获得的公开密钥证书。

```
userCertificate                 ATTRIBUTE     ::=     {
    WITH SYNTAX                 Certificate
    EQUALITY MATCHING RULE      certificateExactMatch
    ID                          id-at-userCertificate}
```

11.2.2 CA证书属性

CA 号码簿条目的 **cACertificate** 属性将用于保存自发放的证书（如果有的话），以及由与本 CA 相同的领域中 CA 发放给本 CA 的证书。在 v3 证书的情况下，这些证书将包括一个 **basicConstraints** 扩展，其 **cA** 值设为 **TRUE**。关于领域的定义纯粹是一个本地问题。

```
cACertificate                   ATTRIBUTE     ::=     {
    WITH SYNTAX                 Certificate
    EQUALITY MATCHING RULE      certificateExactMatch
    ID                          id-at-cAcertificate }
```

11.2.3 交叉证书对属性

CA 号码簿条目 **crossCertificatePair** 属性的 **issuedToThisCA** 元素将用于保存所有的证书，除了那些发放给本 CA 的自发放证书。可选地，CA 号码簿条目 **crossCertificatePair** 属性的 **issuedByThisCA** 元素可以包含一个由本 CA 发放给其他 CA 的证书子集。如果一个 CA 向另一个 CA 发放一个证书，并且对象 CA 不从属于层次结构中的发放者 CA，那么发放者 CA 应将该证书置于其自身号码簿条目 **crossCertificatePair** 属性的 **issuedByThisCA** 元素中。当 **issuedToThisCA** 元素和 **issuedByThisCA** 元素二者都出现在一个单个的属性值中时，一个证书中的发放者名称应匹配于另一个证书中的对象名称，反之亦然，并且一个证书中的对象公开密钥应能对另一个证书中的数字签名进行验证，反之亦然。术语 **forward** 用在以往的版本中，用于 **issuedToThisCA**，术语 **reverse** 用在以往的版本中，用于 **issuedByThisCA**。

当一个 **issuedByThisCA** 元素出现时，**issuedToThisCA** 元素值和 **issuedByThisCA** 元素值无需保存在相同的属性值中；换言之，它们可以保存在一个单个的属性值中或保存在两个属性值中。

在 v3 证书的情况下，这些证书将包括一个 **basicConstraints** 扩展，其 **cA** 值设为 **TRUE**。

```
crossCertificatePair                    ATTRIBUTE      ::=      {
    WITH SYNTAX                         CertificatePair
    EQUALITY MATCHING RULE              certificatePairExactMatch
    ID                                  id-at-crossCertificatePair }

CertificatePair             ::=     SEQUENCE {
    issuedToThisCA          [0]     Certificate OPTIONAL,
    issuedByThisCA          [1]     Certificate OPTIONAL
                                        -- 至少应存在一对 --}
(WITH COMPONENTS { ..., issuedToThisCA PRESENT} |
WITH COMPONENTS { ..., issuedByThisCA PRESENT})
```

11.2.4 证书撤消清单属性

以下属性包含一个撤消证书清单。

```
certificateRevocationList               ATTRIBUTE      ::=      {
    WITH SYNTAX                         CertificateList
    EQUALITY MATCHING RULE              certificateListExactMatch
    ID                                  id-at-certificateRevocationList }
```

11.2.5 机构撤消清单属性

以下属性包含一个撤消的机构证书清单。

```
authorityRevocationList                 ATTRIBUTE      ::=      {
    WITH SYNTAX                         CertificateList
    EQUALITY MATCHING RULE              certificateListExactMatch
    ID                                  id-at-authorityRevocationList }
```

11.2.6 Delta撤消清单属性

定义以下属性类型用于持有号码簿条目中的一个 dCRL：

```
deltaRevocationList                     ATTRIBUTE ::= {
    WITH SYNTAX                         CertificateList
    EQUALITY MATCHING RULE              certificateListExactMatch
    ID                                  id-at-deltaRevocationList }
```

11.2.7 支持的算法属性

定义一个号码簿属性是为了为算法选择提供支持，以便在与利用本号码簿规范中定义的证书的远程终端实体进行通信时使用。以下 ASN.1 定义了本（多值）属性：

```
supportedAlgorithms ATTRIBUTE ::= {
    WITH SYNTAX                     SupportedAlgorithm
    EQUALITY MATCHING RULE          algorithmIdentifierMatch
    ID                              id-at-supportedAlgorithms }
```

```
SupportedAlgorithm ::= SEQUENCE {
    algorithmIdentifier                 AlgorithmIdentifier,
    intendedUsage               [0]     KeyUsage OPTIONAL,
    intendedCertificatePolicies [1]     CertificatePoliciesSyntax OPTIONAL }
```

多值属性的每个值都将拥有一个独特的 **algorithmIdentifier** 值。**intendedUsage** 部件的值指明了算法的计划用法（请参见有关已被认可用法的第 8.2.2.3 节）。**intendedCertificatePolicies** 部件的值用于确定证书策略，以及可选地，已确定算法可以使用的证书策略限定符。

11.2.8 认证实施声明属性

certificationPracticeStmt 属性用于保存有关某个机构认证实施声明的信息。

```
certificationPracticeStmt   ATTRIBUTE     ::=     {
    WITH SYNTAX             InfoSyntax
    ID                      id-at-certificationPracticeStmt }

InfoSyntax          ::=         CHOICE {
    content         DirectoryString {ub-content},
    pointer         SEQUENCE {
        name                    GeneralNames,
        hash                    HASH { HashedPolicyInfo } OPTIONAL } }

POLICY    ::=   TYPE-IDENTIFIER

HashedPolicyInfo        ::=     POLICY.&Type( {Policies} )

Policies POLICY ::= {...} -- 由实施者来定义 --
```

如果 **content** 出现，那么包括机构认证实施声明的完整内容。

如果 **pointer** 出现，那么 **name** 部件参考一个或多个位置，可以在这些位置上找到机构认证实施声明的拷贝。如果 **hash** 部件出现，那么它将包含一个有关认证实施声明内容的散列，应该在所参考的位置上能找到它。该散列可用于对所参考的文档进行完整性检查。

11.2.9 证书策略属性

certificatePolicy 属性用于保存有关证书策略的信息。

```
certificatePolicy           ATTRIBUTE     ::=     {
    WITH SYNTAX             PolicySyntax
    ID                      id-at-certificatePolicy }

PolicySyntax    ::=         SEQUENCE {
    policyIdentifier        PolicyID,
    policySyntax            InfoSyntax
}

PolicyID    ::=     CertPolicyId
```

policyIdentifier 部件包括注册用于特殊证书策略的对象标识符。

如果 **content** 出现，那么包括证书策略的完整内容。

如果 **pointer** 出现，那么 **name** 部件参考一个或多个位置，可以在这些位置上找到证书策略的拷贝。如果 **hash** 部件出现，那么它将包含一个有关证书策略内容的散列，应该在所参考的位置上能找到它。该散列可用于对所参考的文档进行完整性检查。

注 — 在本属性中包括一个散列的选项纯粹为了对来自数据源而非号码簿的数据进行完整性检查。需要对号码簿中保存的散列做好保护。号码簿安全服务，包括强鉴权、访问控制与/或已签署的属性可以用于此目的。另外，即使散列匹配于最初的 CP/CPS 文档，也需要额外的安全措施来确保最初的规范自身是正确的文档（例如，文档经某个适当的机构签署）。

11.2.10 PKI通路属性

PKI 通路属性用于保存认证通路，每个由一系列证书组成。

```
pkiPath     ATTRIBUTE       ::= {
    WITH SYNTAX             PkiPath
    ID                      id-at-pkiPath }
```

本属性可以保存在对象类别 **pkiCA** 或 **pkiUser** 的号码簿条目中。

当保存在 **pkiCA** 条目中时，本属性的值包含认证通路，这些认证通路不包括终端实体证书。这样，属性用于保存认证通路，与该 CA 相关的信赖方常用这些认证通路。本属性的值可以与属性值中最后一个证书对象发放的任何终端实体证书结合使用。

当保存在 **pkiUser** 条目中时，本属性的值包含认证通路，这些认证通路包括终端实体证书。在这种情况下，终端实体为以下用户，即其条目持有属性。属性值代表发放给该用户的证书的完整认证通路。

11.3 PKI号码簿匹配规则

本号码簿规范分别定义了用于 **Certificate**、**CertificatePair**、**CertificateList**、**CertificatePolicy** 和 **SupportedAlgorithm** 类型属性的匹配规则。本节还定义了便于从持有多个证书或 CRL 的多值属性中选择带有特殊特性的证书或 CRL 的匹配规则。增强的证书匹配规则提供了对号码簿条目中所持有的证书进行更复杂匹配的能力。

11.3.1 证书准确匹配

证书准确匹配规则对出现的值是否与 **Certificate** 类型的属性值相同进行比较。它惟一地选择一个单一证书。

```
certificateExactMatch MATCHING-RULE ::= {
    SYNTAX      CertificateExactAssertion
    ID          id-mr-certificateExactMatch }

CertificateExactAssertion ::= SEQUENCE {
    serialNumber    CertificateSerialNumber,
    issuer          Name }
```

如果属性值中的各部件匹配出现值中的各部件，那么本匹配规则返回 TRUE。

11.3.2 证书匹配

证书匹配规则对出现的值是否与 **Certificate** 类型的属性值相同进行比较。根据不同特性，它选择一个或多个证书。

```
certificateMatch MATCHING-RULE ::= {
    SYNTAX      CertificateAssertion
    ID          id-mr-certificateMatch }

CertificateAssertion ::= SEQUENCE {
    serialNumber            [0]     CertificateSerialNumber     OPTIONAL,
    issuer                  [1]     Name                        OPTIONAL,
    subjectKeyIdentifier    [2]     SubjectKeyIdentifier        OPTIONAL,
    authorityKeyIdentifier  [3]     AuthorityKeyIdentifier      OPTIONAL,
    certificateValid        [4]     Time                        OPTIONAL,
    privateKeyValid         [5]     GeneralizedTime             OPTIONAL,
    subjectPublicKeyAlgID   [6]     OBJECT IDENTIFIER           OPTIONAL,
    keyUsage                [7]     KeyUsage                    OPTIONAL,
    subjectAltName          [8]     AltNameType                 OPTIONAL,
    policy                  [9]     CertPolicySet               OPTIONAL,
    pathToName              [10]    Name                        OPTIONAL,
    subject                 [11]    Name                        OPTIONAL,
    nameConstraints         [12]    NameConstraintsSyntax       OPTIONAL
}

AltNameType ::= CHOICE {
    builtinNameForm     ENUMERATED {
                            rfc822Name              (1),
                            dNSName                 (2),
                            x400Address             (3),
                            directoryName           (4),
                            ediPartyName            (5),
                            uniformResourceIdentifier (6),
                            iPAddress               (7),
                            registeredId            (8) },
    otherNameForm       OBJECT IDENTIFIER }

CertPolicySet ::= SEQUENCE SIZE (1..MAX) OF CertPolicyId
```

如果出现在出现值中的所有部件都匹配于属性值的对应部件，那么本匹配规则返回 TRUE，如下所示：

如果属性值中本部件的值等于出现值中的值，那么 **serialNumber** 匹配；

如果属性值中本部件的值等于出现值中的值，那么 **issuer** 匹配；

如果保存属性值中本部件的值等于出现值中的值，那么 **subjectKeyIdentifier** 匹配；如果保存属性值不包含任何对象密钥标识符扩展，那么 **subjectKeyIdentifier** 不匹配；

如果保存属性值中本部件的值等于出现值中的值，那么 **authorityKeyIdentifier** 匹配；如果保存属性值不包含任何机构密钥标识符扩展，或者如果出现值中的所有部件不是都出现在保存属性值中，那么 **authorityKeyIdentifier** 不匹配；

如果出现值落于保存属性值的有效周期内，那么 **certificateValid** 匹配；

如果出现值落于保存属性值专用密钥使用周期扩展所指明的周期内，或者如果在保存属性值中没有任何专用密钥使用周期扩展，那么 **privateKeyValid** 匹配；

如果它等于保存属性值 **subjectPublicKeyInformation** 部件 **algorithmIdentifier** 的 **algorithm** 部件，那么 **subjectPublicKeyAlgID** 不匹配；

如果出现值中设置的所有位也都在保存属性值的密钥使用扩展中进行了设置，或者在保存属性值中没有任何密钥使用扩展，那么 **keyUsage** 匹配；

如果保存属性值包含带名称类型相同的 **AltNames** 部件的对象可选名称扩展，类型如出现值中所示，那么 **subjectAltName** 匹配；

如果至少一个出现的 **CertPolicySet** 成员出现在保存属性值的证书策略扩展中，或者如果出现证书或保存证书包含 **policy** 部件中的特殊值 **anyPolicy**，那么 **policy** 匹配；如果在保存属性值中没有任何证书策略，那么 **policy** 不匹配；

除非证书有一个名称约束扩展，它禁止对出现的名称值构建一条认证通路，否则 **pathToName** 匹配；

如果属性值中本部件的值等于出现值中的值，那么 **subject** 匹配；

如果保存属性值中的对象名称在出现值许可子树部件值所给出的名称空间内，并且不在出现值排除在外子树部件值所给出的名称空间内，那么 **nameConstraints** 匹配。

11.3.3 证书对准确匹配

证书对准确匹配规则对出现的值是否与 **Certificate** 类型的属性值相同进行比较。它惟一地选择一个单一交叉证书对。

```
certificatePairExactMatch MATCHING-RULE          ::=      {
    SYNTAX      CertificatePairExactAssertion
    ID          id-mr-certificatePairExactMatch }

CertificatePairExactAssertion ::= SEQUENCE {
    issuedToThisCAAssertion         [0] CertificateExactAssertion OPTIONAL,
    issuedByThisCAAssertion         [1] CertificateExactAssertion OPTIONAL }
    ( WITH COMPONENTS                   {..., issuedToThisCAAssertion PRESENT} |
      WITH COMPONENTS                   {..., issuedByThisCAAssertion PRESENT} )
```

如果出现在出现值 **issuedToThisCAAssertion** 和 **issuedByThisCAAssertion** 部件中的部件分别匹配于保存属性值 **issuedToThisCA** 和 **issuedByThisCA** 部件中的对应部件，那么本匹配规则返回 TRUE。

11.3.4 证书对匹配

证书对匹配规则对出现的值与类型 **CertificatePair** 的属性值进行比较。它基于对中 **issuedToThisCA** 或 **issuedByThisCA** 证书不同的特性，选择一个或多个交叉证书对。

```
certificatePairMatch MATCHING-RULE ::=     {
    SYNTAX      CertificatePairAssertion
    ID          id-mr-certificatePairMatch }

CertificatePairAssertion ::= SEQUENCE {
    issuedToThisCAAssertion     [0] CertificateAssertion OPTIONAL,
    issuedByThisCAAssertion     [1] CertificateAssertion OPTIONAL }
```

```
    ( WITH COMPONENTS          {..., issuedToThisCAAssertion PRESENT} |
    WITH COMPONENTS            {..., issuedByThisCAAssertion PRESENT} )
```

如果出现在出现值 **issuedToThisCAAssertion** 和 **issuedByThisCAAssertion** 部件中的所有部件都分别匹配于保存属性值 **issuedToThisCA** 和 **issuedByThisCA** 部件中的对应部件，那么本匹配规则返回 TRUE。

11.3.5 证书清单准确匹配

证书清单准确匹配规则对出现的值与类型 **CertificateList** 的属性值进行比较，看它们是否相同。它惟一地选择一个单个的 CRL。

```
certificateListExactMatch MATCHING-RULE ::=        {
    SYNTAX      CertificateListExactAssertion
    ID          id-mr-certificateListExactMatch }

CertificateListExactAssertion ::= SEQUENCE {
    issuer                  Name,
    thisUpdate              Time,
    distributionPoint       DistributionPointName OPTIONAL }
```

如果保存属性值中的各部件匹配于出现值中的各部件，那么规则返回 TRUE。如果 **distributionPoint** 部件出现，那么它应至少以一种名称形式匹配。

11.3.6 证书清单匹配

证书清单匹配规则对出现的值与类型 **CertificateList** 的属性值进行比较。它基于不同的特性，选择一个或多个 CRL。

```
certificateListMatch MATCHING-RULE ::= {
    SYNTAX      CertificateListAssertion
    ID          id-mr-certificateListMatch }

CertificateListAssertion ::= SEQUENCE {
    issuer                          Name                    OPTIONAL,
    minCRLNumber            [0]     CRLNumber               OPTIONAL,
    maxCRLNumber            [1]     CRLNumber               OPTIONAL,
    reasonFlags                     ReasonFlags             OPTIONAL,
    dateAndTime                     Time                    OPTIONAL,
    distributionPoint       [2]     DistributionPointName   OPTIONAL,
    authorityKeyIdentifier  [3]     AuthorityKeyIdentifier  OPTIONAL }
```

如果出现在出现值中的所有部件都匹配于保存属性值的对应部件，那么本匹配规则返回 TRUE，如下所示：

如果属性值中本部件的值等于出现值中的值，那么 **issuer** 匹配；

如果其值小于或等于保存属性值 CRL 号码扩展中的值，那么 **minCRLNumber** 匹配；如果保存属性值不包含任何 CRL 号码扩展，那么 **minCRLNumber** 不匹配；

如果其值大于或等于保存属性值 CRL 号码扩展中的值，那么 **maxCRLNumber** 匹配；如果保存属性值不包含任何 CRL 号码扩展，那么 **maxCRLNumber** 不匹配；

如果出现值中设置的任何位还在保存属性值发放分发点扩展的 **onlySomeReasons** 部件中进行了设置，那么 **reasonFlags** 匹配；如果保存属性值在发放分发点扩展中不包含任何 **reasonFlags**，或者如果保存属性值不包含任何发放分发点扩展，那么 **reasonFlags** 也匹配；

注 — 即使 CRL 匹配于 **reasonFlags** 的某个特殊值，CRL 仍不可以包含任何带该理由代码的撤消通知。

如果值等于或晚于保存属性值 **thisUpdate** 部件中的值，并且早于保存属性值 **nextUpdate** 部件中的值，那么 **dateAndTime** 匹配；如果保存属性值不包含任何 **nextUpdate** 部件，那么 **dateAndTime** 不匹配；

如果保存属性值包含一个发放分发点扩展，并且出现值中本部件的值等于该扩展中至少一个名称形式中的对应值，那么 **distributionPoint** 匹配；

如果保存属性值中本部件的值等于出现值中的值，那么 **authorityKeyIdentifier** 匹配；如果保存属性值不包含任何机构密钥标识符扩展，或者如果出现值中的所有部件不是都出现在保存属性值中，那么 **authorityKeyIdentifier** 不匹配。

11.3.7 算法标识符匹配

算法标识符匹配规则对出现的值与类型 **SupportedAlgorithms** 的属性值进行比较，看它们是否相同。

```
algorithmIdentifierMatch MATCHING-RULE   ::=  {
    SYNTAX      AlgorithmIdentifier
    ID          id-mr-algorithmIdentifierMatch }
```

如果出现的值等同于保存属性值的 **algorithmIdentifier** 部件，那么规则返回 TRUE。

11.3.8 策略匹配

策略匹配规则对出现的值与类型 **CertificatePolicy** 的属性值或类型 **privPolicy** 的属性值进行比较，看它们是否相同。

```
policyMatch MATCHING-RULE        ::= {
    SYNTAX      PolicyID
    ID          id-mr-policyMatch }
```

如果出现的值等同于保存属性值的 **policyIdentifier** 部件，那么规则返回 TRUE。

11.3.9 PKI通路匹配

pkiPathMatch 匹配规则对出现的值与类型 **pkiPath** 的属性值进行比较，看它们是否相同。使用证书的系统可以使用本匹配规则来选择一条开始于其信任的 CA 发放的证书的通路，并结束于发放给特定对象的证书。

```
pkiPathMatch MATCHING-RULE  ::= {
    SYNTAX      PkiPathMatchSyntax
    ID          id-mr-pkiPathMatch }

PkiPathMatchSyntax      ::=  SEQUENCE {
    firstIssuer         Name,
    lastSubject         Name }
```

如果 **firstIssuer** 部件中出现的值匹配于保存值中 **SEQUENCE** 中第一个证书的 **issuer** 字段的对应元素，以及 **lastSubject** 部件中出现的值匹配于保存值中 **SEQUENCE** 中最后一个证书的对象字段的对应元素，那么本匹配规则返回 TRUE。如果二者都匹配失败，那么本匹配规则返回 FALSE。

11.3.10 增强的证书匹配

增强的证书匹配规则对出现的值与类型 **Certificate** 的属性值进行比较。它基于不同的特性，选择一个或多个证书。

```
enhancedCertificateMatch MATCHING-RULE  ::= {
    SYNTAX      EnhancedCertificateAssertion
    ID          id-mr-enhancedCertificateMatch }

EnhancedCertificateAssertion  ::= SEQUENCE {
    serialNumber            [0]   CertificateSerialNumber     OPTIONAL,
    issuer                  [1]   Name                        OPTIONAL,
    subjectKeyIdentifier    [2]   SubjectKeyIdentifier        OPTIONAL,
    authorityKeyIdentifier  [3]   AuthorityKeyIdentifier      OPTIONAL,
    certificateValid        [4]   Time                        OPTIONAL,
    privateKeyValid         [5]   GeneralizedTime             OPTIONAL,
    subjectPublicKeyAlgID   [6]   OBJECT IDENTIFIER           OPTIONAL,
    keyUsage                [7]   KeyUsage                    OPTIONAL,
    subjectAltName          [8]   AltName                     OPTIONAL,
    policy                  [9]   CertPolicySet               OPTIONAL,
    pathToName              [10]  GeneralNames                OPTIONAL,
    subject                 [11]  Name                        OPTIONAL,
    nameConstraints         [12]  NameConstraintsSyntax       OPTIONAL
}

    (ALL EXCEPT ({-- 无；至少应出现一个部件。 --}))

    AltName  ::= SEQUENCE {
        altnameType     AltNameType,
        altNameValue    GeneralName OPTIONAL }
```

号码簿搜索操作允许多个 **EnhancedCertificateAssertion** 值结合进过滤器规范中，包括与/或逻辑。如果出现在出现值中的所有部件都匹配于属性值的对应部件，那么本匹配规则返回 TRUE，如下所示：

关于 **serialNumber**、**issuer**、**subjectKeyIdentifier**、**authorityKeyIdentifier**、**certificateValid**、**privateKeyValid**、**policy**、**subject** 和 **nameConstraints** 部件的匹配是为 **certificateMatch** 匹配规则中的相同部件定义的。

subjectAltName 部件包含一个 **altNameType** 字段，以及可选的 **altNameValue** 字段。如果 **altNameValue** 出现，那么值的名称形式应与 **altNameType** 中所述的名称形式相同。

如果至少一个以下条件为 TRUE，那么 **subjectAltName** 匹配：

— 出现值只包含 **altNameType** 部件，保存属性值包含带类型相同的 **AltNames** 部件的对象可选名称扩展，如出现值中所示；

— 出现值包含 **altNameType** 和 **altNameValue** 部件，保存属性值包含带类型和值相同的 **AltNames** 部件的对象可选名称扩展，如出现值中所示。

如果至少一个以下条件为 TRUE，那么 **subjectAltName** 匹配失败：

— 保存属性值不包含对象可选名称扩展；

— 保存属性值包含对象可选名称扩展，但 **AltNames** 部件不包括如出现值中所确定的相同类型；

— 出现值包含 **altNameType** 和 **altNameValue** 部件，保存属性值包含带类型相同的 **AltNames** 部件的对象可选名称扩展，类型如出现值中所示，但保存值不包含出现值中所示类型的相同值。

如果出现值包含 **altNameType** 和 **altNameValue** 部件，并且保存属性值包含带类型相同的 **AltNames** 部件的对象可选名称扩展，类型如出现值中所示（但对该类型，号码簿无法为确定匹配与否而对其值进行比较），那么 **subjectAltName** 匹配是不明确的。这可能是由于名称形式不适合匹配，或者是由于号码簿无法执行要求的比较。

除非证书有一个名称约束扩展，它禁止对任何出现的名称值构建一条认证通路，否则 **pathToName** 匹配。例如，如果尝试对证书进行检索，这些证书形成一条对用户证书的通路，其对象值为“dc=com; dc=corporate; cn=john.smith”，那么在搜索操作中包括一个声明可能会有用，搜索操作在 **pathToName** 部件中包含本 DN。包含名称约束扩展的保存证书（不包括完整子树的下基“dc=com; dc=company A”）将无法完成对该用户证书的认证通路验证，并因此无法成为本样本声明的匹配值。

第 3 部分 — 属性证书框架

此处定义的属性证书框架为构建特权管理基础设施（PMI）奠定了基础。这些基础设施能为诸如访问控制这样的应用提供支持。

通过一个称为属性证书的数字签署的数据结构，或者通过一个包含扩展（为此目的而明确定义）的公开密钥证书，机构可以将一个特权绑定于一个实体。属性证书的格式在此定义，包括一个扩展性机制和一系列特殊的证书扩展。可能需要也可能不需要属性证书的撤消。例如，在某些环境中，属性证书的有效期可能非常短（例如，几分钟），使得无需撤消方案。如果出于某种原因机构撤消一个先前发放的属性证书，那么用户需要能够知道撤消已经发生，从而使之不使用一个不可信赖的证书。撤消清单是一种可用于通知用户有关撤消情况的方案。撤消清单的格式在本规范的第 2 部分中定义，包括一个扩展性机制和一系列撤消清单扩展。此处还定义了额外的扩展。在证书和撤消清单这两种情况下，其他团体还可以定义额外的、对其特定环境有用的扩展。

在对应用使用证书之前，使用系统的属性证书需要对该证书进行确认。执行确认任务的程序也在此处定义，包括对证书本身的完整性、其撤消状态、其对计划用途的有效性进行验证。

本框架包括众多可选元素，它们只在某些环境中是合适的。虽然完整定义了模型，但本框架可以在以下环境中使用，即所定义模型的所有部件不是都使用。例如，存在无需撤消属性证书的环境。特权委托和角色使用也是

本框架涉及的问题，它们不是普遍适用的。不过，在本规范中包括了这些内容，因此也可以为那些需要这些内容的环境提供支持。

号码簿使用属性证书来对号码簿信息提供基于规则的控制。

12 属性证书

公开密钥证书原则上用于提供身份服务，基于它，可以建立其他安全服务，例如数据完整性、实体鉴权、机密性和授权。在本规范中提供了两种不同的、将一个特权属性绑定于一个持有者的机制。

如果通过发放 CA 行为实现了特权与对象的关联，那么与实体鉴权服务结合使用的公开密钥证书可以直接提供鉴权服务。公开密钥证书可以包含一个 **subjectDirectoryAttributes** 扩展，扩展包含与公开密钥证书对象相关联的特权。本机制在以下情况下是合适的，即发放公开密钥证书（CA）的机构也是委托特权的机构（AA），以及特权的有效期对应公开密钥证书的有效期。终端实体不得作为 AA。如果在公开密钥证书中包括任何第 15 节中所定义的扩展，那么这些扩展同样适用于在该公开密钥证书 **subjectDirectoryAttributes** 扩展中指派的所有特权。

在更普遍的情况下，实体特权将拥有不与公开密钥证书有效期相匹配的寿命。特权常拥有一个更短的寿命。指派特权的机构往往不同于为同一实体发放公开密钥证书的机构，不同的特权可以由不同的属性机构（AA）来指派。还可以临时指派特权，特权的“开/关”完全可异步于公开密钥证书的寿命与/或异步于不同 AA 发放的实体特权。AA 发放的属性证书的使用提供了一个灵活的特权管理基础设施（PMI），其建立和管理可以独立于 PKI。同时，二者之间存在一种关系，据此可以利用 PKI 来鉴别属性证书中发放者和持有者的身份。

12.1 属性证书结构

属性证书是一个独立于对象公开密钥证书的结构。一个对象可以有多个与其每个公开密钥证书相关的属性证书。不要求相同的机构为一个用户既创建公开密钥证书，又创建属性证书；事实上，常常另行规定分开职责。在由不同的机构负责发放公开密钥证书和属性证书的环境中，将利用不同的专用签署密钥，来对由认证机构（CA）发放的公开密钥证书和由属性机构（AA）发放的属性证书进行签署。在一个单个实体既是发放公开密钥证书的 CA 又是发放属性证书的 AA 的环境中，强烈建议用于签署属性证书的密钥不同于签署公开密钥证书的密钥。发放机构与接收证书实体之间的交流不在本规范的讨论范围之内。

属性证书定义如下：

```
AttributeCertificate ::= SIGNED {AttributeCertificateInfo}

AttributeCertificateInfo ::= SEQUENCE
    {
    version                     AttCertVersion, -- 版本为 v2
    holder                      Holder,
    issuer                      AttCertIssuer,
    signature                   AlgorithmIdentifier,
    serialNumber                CertificateSerialNumber,
    attrCertValidityPeriod      AttCertValidityPeriod,
    attributes                  SEQUENCE OF Attribute,
    issuerUniqueID              UniqueIdentifier OPTIONAL,
    extensions                  Extensions       OPTIONAL
    }

AttCertVersion  ::= INTEGER { v2(1) }

Holder      ::=  SEQUENCE
                 {
                 baseCertificateID          [0] IssuerSerial            OPTIONAL,
```

```
        -- 持有者公开密钥证书的发布者和序列号
    entityName              [1] GeneralNames        OPTIONAL,
        -- 实体或角色的名称
    objectDigestInfo        [2] ObjectDigestInfo    OPTIONAL
        -- 用于直接鉴别持有者，例如，一个可执行者
        -- baseCertificateID、entityName 或 objectDigestInfo 中至少应出现一个 -- }

ObjectDigestInfo  ::= SEQUENCE {
    digestedObjectType  ENUMERATED {
      publicKey             (0),
      publicKeyCert         (1),
      otherObjectTypes      (2) },
    otherObjectTypeID       OBJECT IDENTIFIER  OPTIONAL,
    digestAlgorithm         AlgorithmIdentifier,
    objectDigest            BIT STRING }

AttCertIssuer ::=     [0]   SEQUENCE {
issuerName                      GeneralNames OPTIONAL,
baseCertificateID           [0] IssuerSerial OPTIONAL,
objectDigestInfo            [1] ObjectDigestInfo OPTIONAL }
--至少应出现一个部件。
  ( WITH COMPONENTS { ..., issuerName PRESENT } |
  WITH COMPONENTS { ..., baseCertificateID PRESENT } |
  WITH COMPONENTS { ..., objectDigestInfo PRESENT } )

IssuerSerial  ::=  SEQUENCE {
    issuer          GeneralNames,
    serial          CertificateSerialNumber,
    issuerUID   UniqueIdentifier OPTIONAL }

AttCertValidityPeriod  ::= SEQUENCE {
    notBeforeTime       GeneralizedTime,
    notAfterTime        GeneralizedTime }
```

在不同的属性证书版本之间，**version** 是有区别的。对依据本规范中的语法进行发放的各属性证书，**version** 为 **v2**。

Holder 字段用于传达属性证书持有者的身份。

如果出现，那么 **baseCertificateID** 部件确定一个特殊的公开密钥证书，在利用该属性证书声明特权时，将用它来鉴别该持有者的身份。

如果出现，那么 **entityName** 部件确定一个或多个持有者的名称。如果 **entityName** 为出现在 **holder** 中的惟一部件，那么在利用该属性证书声明特权时，任何拥有其中一个名称的公开密钥证书其对象都可用于鉴别该持有者的身份。如果 **baseCertificateID** 和 **entityName** 二者都出现，那么只能使用由 **baseCertificateID** 规定的证书。在这种情况下，包括的 **entityName** 只能作为一个工具，来帮助特权验证者定位已确定的公开密钥证书。

注 1 — 单独使用 **GeneralNames** 来确定持有者存在一定的风险，这指的只是持有者的名称。出于向该持有者发放特权的目的，通常这是不足以实现对持有者身份的鉴别的。不过，使用发放者的名称和某个特定公开密钥证书的序列号，使属性证书的发放者能够在发放特殊的公开密钥证书时依赖 CA 执行的鉴权过程。另外，**GeneralNames** 中的某些选项（例如，**IPAddress**）是不适合用于命名属性证书持有者的，尤其在持有者是一个角色而不是一个单独的实体时。**GeneralNames** 单独作为持有者标识符的另一个问题是，该构件中的许多名称形式没有严格的注册机构或名称指派过程。

如果出现，那么 **objectDigestInfo** 部件直接用于鉴别持有者的身份，包括可执行的持有者（例如，一个 Java 小程序）。通过比较对应信息的摘要来鉴别持有者，它由特权验证者使用在 **objectDigestInfo** 中、利用 **objectDigest** 内容确定的同一算法来创建。如果二者相同，那么出于使用该属性证书声明特权的目的，对持有者进行鉴别。

— 当包括一个实体公开密钥散列时，应指明 **publicKey**。散列一个公开密钥不能惟一地确定一个证书（即相同的密钥值可以出现在多个证书中）。为了将一个属性证书连接至一个公开密钥，利用该公开密钥的表达式来计算散列，公开密钥将出现在一个公开密钥证书中。特别地，散列算法的输入

将为密钥 **SubjectPublicKeyInfo** 表达式的 DER 编码。注意：这包括 **AlgorithmIdentifier** 以及 **BIT STRING**。注意：如果已从公开密钥证书中抽取作为散列函数输入的公开密钥值，那么有可能（例如，若继承了数字签名算法的参数），则它可能不是散列的充分输入。这种情况下，散列的正确输入将包括已集成参数的值，因此可能不同于出现在公开证书中的 **SubjectPublicKeyInfo**。

— 当散列公开密钥证书时，应指明 **publicKeyCert**；散列针对公开密钥证书的整个 DER 编码，包括各签名位。

— 当散列不是为公开密钥或公开密钥证书的对象（例如，软件对象）时，应指明 **otherObjectTypes**。对象类型的身份可以任意提供。被散列的那部分对象可以通过明确声明的类型标识符来确定，或者如果未提供标识符，那么通过对象的使用范围来确定。

issuer 字段用于传达 AA 的身份，AA 负责发放证书。

— 如果出现，那么 **issuerName** 部件确定一个或多个发放者的名称。

— 如果出现，那么通过参考某个特定的公开密钥证书，**baseCertificateID** 部件确定发放者，对该证书，发放者为对象。

— 如果出现，那么通过为发放者提供一个识别信息散列，**objectDigestInfo** 部件确定发放者。

signature 确定用于数字签署属性证书的密码算法。

serialNumber 为序列号，它惟一确定其发放者范围内的属性证书。

attrCertValidityPeriod 字段用于传达时间周期，在此期间，认为属性证书是有效的，时间周期以 **GeneralizedTime** 格式进行表示。

attributes 字段包含与持有者相关的属性，该持有者正在接受认证（例如，特权）。

注 2 — 在属性描述符属性证书的情况下，该属性序列可以为空。

在发放者部件不够充分的情况下，可以使用 **issuerUniqueID** 来确定属性证书的发放者。

extensions 字段允许向属性证书增加新的字段。

如果在扩展中出现了未知的元素，并且扩展未被标记为关键的，那么将依据 ITU-T X.519 建议书 | ISO/IEC 9594-5 第 12.2.2 节中所述的扩展性规则，忽略这些未知的元素。

本部分中所述的属性证书框架主要集中于模型，在该模型中，特权置于属性证书中。不过，如前所述，在本部分中定义的证书扩展也可以置于使用 **subjectDirectoryAttributes** 扩展的公开密钥证书中。

12.2 属性认证通路

恰如公开密钥证书，可能需要传达一条属性证书通路（例如，在一个用于声明特权的应用协议内）。以下 ASN.1 数据类型可用于描述一条属性证书通路：

```
AttributeCertificationPath ::= SEQUENCE {
    attributeCertificate        AttributeCertificate,
    acPath                      SEQUENCE OF ACPathData OPTIONAL }

ACPathData ::= SEQUENCE {
    certificate             [0] Certificate OPTIONAL,
    attributeCertificate    [1] AttributeCertificate OPTIONAL }
```

13 属性机构、SOA和证书机构关系

属性机构（AA）和认证机构（CA）逻辑上（以及在许多情况下，物理上）是完全独立的。“身份”的创建和维护可以（常常应该）独立于 PMI。因此，整个 PKI，包括 CA，在 PMI 建立之前，就可以是存在和可操作的。虽然 CA 为其域内身份的机构源，但它不是自动地为特权的机构源。因此，CA 自身不必是一个 AA，以及在逻辑意义上不必对决定负责，从而使其他实体能够充当 AA（例如，通过包括在其身份证书中包括这样一个指定）。

机构源（SOA）是特权验证者信任的实体，它是最终负责指派一系列特权的实体。通过向某些 SOA 托付特定的功能（例如，一个托付读特权，另一个托付写特权），某个资源可以限制 SOA 的权力。SOA 本身是一个 AA，它向其他实体发放证书，在证书中特权被指派给这些实体。SOA 类似于 PKI 中的一个“根 CA”或“信任锚点”，特权验证者在其中托付由 SOA 签署的各证书。在某些环境中，需要 CA 对可以作为 SOA 的实体实施严格控制。该框架提供了一种支持此要求的机制。在其他环境中，不需要这种控制，用于确定在此类环境中可以作为 SOA 的实体的机制不在本规范的讨论范围之内。

本框架是灵活的，能够满足众多环境类型的要求。

a） 在许多环境中，所有特权将通过一个单个 AA 直接指派给单独的实体，即 SOA。

b） 其他环境可能需要支持可选的角色特性，据此将证书发放给各个体，证书将各种不同的角色指派给它们。与角色相关的特权隐含地指派给这些个体。角色特权本身在发放给角色自身的属性证书中指派，或者通过某些其他方式指派（例如，在本地配置）。

c） 本框架的另一个可选特性是支持特权委托。如果进行委托，那么 SOA 向还允许作为 AA 并进一步委托特权的实体指派特权。委托可以通过若干中间 AA 继续进行，直至它最终指派给一个不能再进一步委托该特权的终端实体。中间 AA 可能能够也可能不能够作为其委托之特权的特权声明者。

d） 在某些环境中，同一物理实体既可以作为一个 AA，也可以作为一个 CA。对同一物理实体的这个双重逻辑角色总是出现在以下情况中，即当特权在公开密钥证书的 **subjectDirectoryAttributes** 扩展中传达时。在其他环境中，由不同的物理实体来充当 CA 和 AA。在后一种情况下，利用属性证书而不是公开密钥证书来指派特权。

当属性证书为其发放者和持有者指向公开密钥证书时，使用 PKI 来鉴别持有者（特权声明者）以及验证发放者的数字签名。

在本规范中描述了两个委托模型。在第一个委托模型中，特权委托者是一个 AA，它能够发放证书，以便向其他 AA 委托该特权。第二个模型允许一个独立的委托服务（DS），在当中实体代表另一个 AA（它能够或不能够发放 AC 自身）发放证书。该 DS 自身不能作为该特权的要求者。DS 模型与以下环境尤其相关，即这些环境希望对在其域中委托的特权集保持某种集中式管理。例如，一个或多个执行委托的 DS 服务器的集合，而不是单独的特权持有者，允许利用一个集中式的工具来确定总的、在某个环境中委托的特权集，并因此使得能够对策略和管理决定进行修改。对 DS 服务器可以有两个不同的配置模型。在一个模型中，由 SOA 将特权指派给特权持有者，这些持有者有权将该特权委托给其他持有者。不过，不像发放用于委托特权自身的属性证书，特权持有者请求 DS 代表它们来委托该特权。DS 本身不持有该特权，因此不能作为该特权的要求者；不过，SOA 授权 DS 代表其他特权持有者来发放属性证书。第二个配置模型类似于第一个模型，除了以下例外情况。DS 实际上是一个持有者，向它指派待委托的特权，但 DS 无权作为该特权的要求者，只能作为委托者。在这种情况下，必须在由 SOA 发放给 DS 的 AC 中对 noAssertion 扩展进行设置。DS 定义为一个间接的发放者。

在两个配置模型中，SOA 都向下属 AA 发放属性/特权。而后 AA 请求 DS 向其他持有者发放这些特权属性的子集。在第二个配置模型中，DS 能够检查 AA 是否在 SOA 所设置的整个范围中都进行委托；在第一个配置模型中，DS 不能进行检查，需要信赖方来检查是否正确进行了委托。

13.1 属性证书中的特权

实体可以以两种方式来获得特权：

— 一个 AA 可以通过创建一个属性证书来单方面地将特权指派给一个实体（可能完全出于其自身动机，或在某个第三方请求下）。该证书可以保存在一个可公开访问的知识库中，并在之后可由一个或多个特权验证者进行处理，以便做出授权决定。所有这些都可能发生，而无需实体的知识或外在行动。

— 可选地，一个实体可以请求某个 AA 的特权。一旦创建，那么可以将该证书返回（只能）给请求实体，当请求访问某个受保护的资源时，请求实体将明确提供该证书。

注意：在两个程序中，AA 都需尽力确保为实体指派了该特权。这可能涉及某些带外机制，类似于由 CA 对身份/密钥对绑定进行认证。

在以下任何之一为 TRUE 的环境中，基于 PMI 的属性证书都是合适的：

— 一个不同的实体负责向持有者指派特殊的特权，而不是向同一对象发放公开密钥证书；

— 有许多特权属性需要从不同的机构指派给持有者；

— 特权的寿命不同于持有者公开密钥证书有效性的寿命（一般情况下特权的寿命更短）；或者

— 只有在某些时间间隔期间，特权才是有效的，这些时间间隔异步于用户的公开密钥有效期或其他特权的有效期。

13.2 公开密钥证书中的特权

在某些环境中，特权通过 CA 的行为与对象实现关联。此类特权可以直接置于公开密钥证书中（从而重用大量已经建立的基础设施），而不是发放属性证书。在这些情况下，特权包括在公开密钥证书的 **subjectDirectoryAttributes** 扩展中。

在以下任何之一为 TRUE 的环境中，本机制都是合适的：

— 相同的物理实体作为一个 CA 和一个 AA；

— 特权的寿命与证书中包括的公开密钥的寿命有关；

— 不允许委托特权；或者

— 允许委托，但对任何一个委托，证书中的所有特权（在 **subjectDirectoryAttributes** 扩展中）都具有相同的委托参数，与委托相关的所有扩展均同等地适用于证书中的所有特权。

14 PMI模型

14.1 普通模型

普通特权管理模型由 3 个实体组成：对象、特权声明者和特权验证者。

对象可以是某个受保护的资源，例如在某个访问控制应用中。将受保护的资源称为对象。这种类型对象拥有可被调用的方法（例如，对象可以是一个防火墙，它拥有“允许条目”对象方法；或者对象可以是一个文件系统中的文件，它拥有读、写和执行对象方法）。在本模型中的另一种类型对象可以是一个在不可否认应用中签署的对象。

特权声明者是持有某个特殊特权并对某个特殊使用范畴声明其特权的实体。

特权验证者是确定所声明的特权对特定使用范畴而言是否充分的实体。

通过/不通过特权验证者所做的决定取决于 4 个因素：

— 特权声明者的特权；

— 适当的特权策略；

— 如果相关的话，当前的环境变量；以及

— 如果相关的话，对象方法的敏感性。

特权持有者的特权反映了证书发放者对该持有者的信任等级，特权持有者将坚持这些方面的策略，它们不通过技术方法实施。该特权封装在特权持有者的属性证书中（或其公开密钥证书的 **subjectDirectoryAttributes** 扩

展中），它可以在调用请求中提交给特权验证者，或者可以通过某些其他方法进行发布，如通过号码簿。整理特权通过使用 **Attribute** 构件来完成，它包含一个 **AttributeType** 和一个 **SET OF AttributeValue**。用于规定特权的某些属性类型的语法可以非常简单，如一个单个 **INTEGER** 或一个 **OCTET STRING**。其他属性类型的句法可以更加复杂。在附件 D 中提供了一个例子。

特权策略用于规定认为对某个特定对象方法的敏感性或使用范畴是足够的特权等级。需要对特权策略的完整性和真实性予以保护。对传达策略存在众多可能性。一个极端的想法是根本不真正传达策略，而是只做简单定义，并只保存在本地的特权验证者环境中。另一个极端的想法是某些策略是“通用的”，应传送给系统中的每一个实体，并被系统中的每一个实体所知晓。在这些极端想法之间存在许多渐变。在本规范中定义了用于在号码簿中保存特权策略信息的模式部件。

特权策略用于规定接受某个特定特权集的门限。也就是说，它准确定义了特权验证者应在何时确定某个提交的特权集是“足够的”，以便它可以准许访问（请求的对象、资源、应用等）特权声明者。

在本规范中，有关特权策略定义的句法不是标准化的。附件 D 包含了两个语法例子，它们可用于此目的。不过，这些只是例子而已。任何语法都可以用于此目的，包括明文。不管用于定义特权策略的语法是什么，都将惟一确定特权策略的每个实例。对象标识符用于此目的。

PrivilegePolicy ::= OBJECT IDENTIFIER

如果相关，那么环境变量涉及需要做出通过/不通过决定的那些策略问题（例如，以天计的时间或当前的账目平衡），通过某些本地方法，它们可供特权验证者使用。对环境变量的描述问题完全是一个本地问题。

如果相关，对象发放敏感性可以反映文档的属性或请求处理，如它声称授权的基金所传达的货币价值，或文档内容的机密性。对象方法的敏感性可以在相关的安全标签中或在对象方法所持有的属性证书中予以明确编码，或者它可以在相关数据对象的结构和内容中予以隐含封装。可以用众多不同方法中的一种对它进行编码。例如，可以在 PMI 范围之外，在文档相关的 X.411 标签中，在 EDIFACT 相互交换的各字段中，对它进行编码，或者在特权验证者的应用中对它进行硬编码。可选地，可以在 PMI 之内，在与对象方法相关的属性证书中，对它进行编码。对某些使用范畴，无需使用任何对象方法敏感性。

一个特权验证者与任何特殊的 AA 之间不必存在任何绑定关系。恰如特权持有者可以拥有由许多不同的 AA 发放给它们的属性证书，特权验证者也可以接受由众多 AA 发放的证书（它们在层次结构上不必相互相关），以便准许访问某个特殊资源。

出于众多目的，属性证书框架可用于管理各种不同类型的特权。本规范中所用的术语，如特权声明者、特权验证者等，独立于特殊的应用或使用。

14.1.1 访问控制范畴的PMI

对访问控制有一个标准的框架（ITU-T X.812 建议书 | ISO/IEC 10181-3），它定义了一个专门针对访问控制应用的对应术语集。此处提供了一个本规范中所用的一般术语与访问控制框架中所用的一般术语之间的映射，用于明确本模型与该规范之间的关系。

本规范中的特权声明者将充当访问控制框架中的“发起者”角色。

本规范中的特权验证者将充当访问控制框架中的“访问控制决定函数（ADF）”角色。

本规范中用于声明特权的对象方法将对应访问控制框架中定义的“目标”。

本规范中的环境变量将对应访问控制框架中的“上下文信息”。

在本规范中讨论的特权策略可以包括“访问控制策略”，以及在访问控制框架中定义的“访问控制策略规则”。

本模型允许在一个待保护的现有资源网络上合理地、无缝地重叠 PMI。尤其是，当特权验证者作为敏感对象方法的网关、授权或拒绝请求调用该对象方法时，使得对象能够得到保护，而对对象本身产生很小的影响或不产生任何影响。特权验证者屏蔽所有的请求，只有那些得到了正确授权的请求才能达到适当的对象方法。

14.1.2 认可范畴的PMI

对不可否认有一个标准的框架（ITU-T X.813 建议书 | ISO/IEC 10181-4），它定义了一个专门针对不可否认的对应术语集。此处提供了一个本规范中所用的一般术语与不可否认框架中所用的一般术语之间的映射，用于明确本模型与该规范之间的关系。

本规范中的特权声明者将充当不可否认框架中的“证据对象”或“发起者”角色。

本规范中的特权验证者将充当不可否认框架中的“证据用户”或“接收者”角色。

本规范中用于声明特权的对象方法将对应不可否认框架中定义的“目标”。

本规范中的环境变量将对应不可否认框架中的“产生或验证日期和时间证据”。

在本规范中所讨论的特权策略可以包括不可否认框架中的“不可否认安全策略”。

14.2 控制模型

控制模型用于描述如何控制对敏感对象方法的访问。模型包括五个部件：特权声明者、特权验证者、对象方法、特权策略、环境变量（请参见图 3）。特权声明者具有特权；对象方法具有敏感性。此处所述的技术使得特权验证者能够依据特权策略，对特权声明者访问对象方法实施控制。特权和敏感性二者都可以是多值参数。

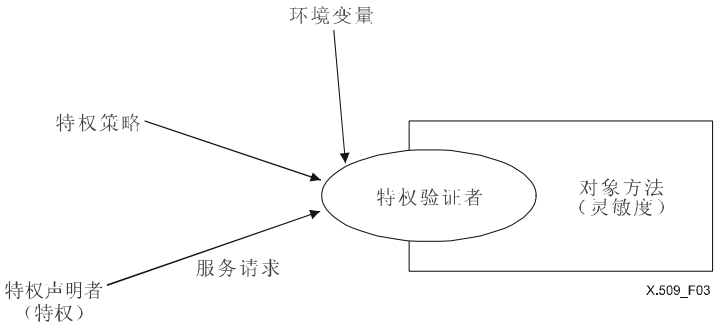


图 3－控制模型

特权声明者可以是一个由公开密钥证书确定的实体，或者是一个由其磁盘镜像摘要确定的可执行对象等。

14.3 委托模型

在某些环境中，可能需要委托特权；不过，这是框架的一个可选项，并不要求所有环境都这样。委托模型包括 4 个部件：特权验证者、SOA、其他 AA 和特权声明者（如图 4 所示）。

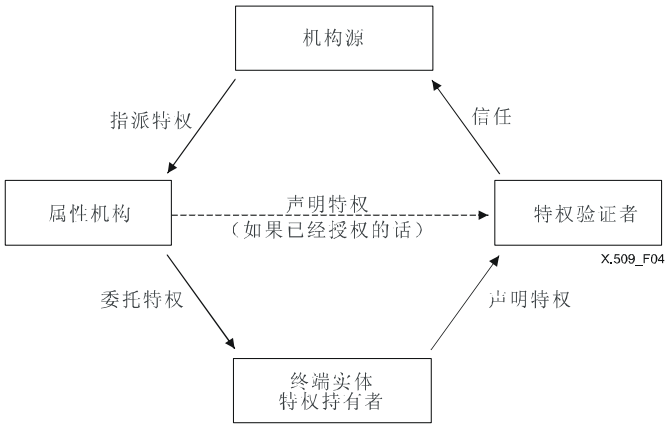


图 4－委托模型

对未使用委托的环境，SOA 是证书的最初发放者，这些证书向特权持有者指派特权。不过，在这种情况下，SOA 授权特权持有者作为 AA，并通过发放包含相同特权（或特权子集）的证书来进一步将该特权委托给其他实体。SOA 可以对能够完成的委托施加约束（例如，限制通路长度，限制能够完成委托的名称空间）。在它发放给进一步特权持有者的证书中，这些中间 AA 中的每一个都可以授权由这些持有者（还可以充当 AA）来完成委托。对委托的一个一般性限制是，任何 AA 都不可以委托比其所持有的特权还要多的特权。委托者还可以进一步限制下行流 AA 的能力。

当使用委托时，特权验证者托付 SOA 将某些或全部这些特权委托给持有者，其中的某些持有者可以进一步将某些或全部这些特权委托给其他持有者。

特权验证者托付 SOA 作为资源某个特定特权集的机构。如果特权声明者的证书不是由该 SOA 发放的，那么特权验证者将确定一条从特权声明者证书到 SOA 发放证书的委托通路。对该委托通路进行确认包括检查每个 AA 是否拥有足够的特权，以及是否拥有适当的权力来委托这些特权。

对特权通过属性证书方式进行传达的情况，委托通路不同于证书确认通路，证书确认通路用于确认委托处理中涉及的各实体的公开密钥证书。不过，公开密钥证书确认处理所提供的真实性质量将与受到保护的对象方法的灵敏性相称。

一条委托通路要么完全由属性证书组成，要么完全由公开密钥证书组成。如果得到授权，在一个属性证书中获得其特权的委托者只能通过发放后续属性证书来委托。同样，如果得到授权，在一个公开密钥证书中获得其特权的委托者只能通过发放后续公开密钥证书来委托。只有 AA 才能委托特权。终端实体不能委托特权。

14.4 角色模型

角色提供了一种间接向各个体指派特权的方法。向个体发放角色指派证书，这些证书通过证书中包含的角色属性向它们指派一个或多个角色。通过角色规范证书向角色名称指派特定的特权，而不是通过属性证书向单个特权持有者指派特权。例如，该层面的间接指派使得能够对指派给角色的特权进行更新，而不影响向各个体指派角色的证书。角色指派证书可以是属性证书或公开密钥证书。角色规范证书可以是属性证书，但不可以是公开密钥证书。如果未使用角色规范证书，那么可以通过其他方法（例如，可以在本地对特权验证者进行配置）来实现向角色指派特权。

以下都是可能的：

— 任何 AA 都可以定义任何数量的角色；

— 可以通过不同的 AA 单独定义和管理角色本身以及角色的各成员；

— 恰如同任何其他特权，可以对角色成员资格进行委托；以及

— 可以为角色和成员资格指派任何适当的寿命。

如果角色指派证书是一个属性证书，那么 **role** 属性包含在属性证书的 **attributes** 部件中。如果角色指派证书是一个公开密钥证书，那么 **role** 属性包含在 **subjectDirectoryAttributes** 部件中。在后一种情况下，包含在公开密钥证书中的任何额外特权都是直接指派给证书对象的特权，而不是指派给角色的特权。

因此，特权声明者可以将角色指派证书提交给特权验证者，证明特权声明者只有一个特殊的角色（例如，“经理”或“购买者”）。由此，特权验证者可以知道或需要通过某些其他方法找到与声明的角色相关的特权，以便做出通过/不通过授权的决定。角色规范证书可用于该目的。

特权验证者需要了解为角色规定的特权。向角色指派这些特权可以在角色规范证书的 PMI 内完成，或在 PMI 外完成（例如，在本地进行配置）。如果在角色规范证书中声明角色特权，那么在本规范中提供以下机制，即实现该证书与特权声明者相关角色指派证书连接的机制。不可以将角色规范证书委托给任何其他实体。角色指派证书的发放者可以独立于角色规范证书的发放者，对这些完全可以单独进行管理（终止、撤消等）。同一证书（属性证书或公开密钥证书）可以是一个角色指派证书，并包含直接针对同一个体的其他特权指派。不过，一个角色规范证书应是一个单独的证书。

注 — 授权框架内角色的使用可以提高通路处理的复杂性，原因是，该功能本质上定义了另一条需要跟随的委托通路。角色指派证书的委托通路可能涉及不同的 AA，并可能独立于发放角色规范证书的 AA。

14.4.1 角色属性

特权属性类型规范通常是一个应用特定的问题，它不在本规范的讨论范围之内。本属性的一个例外是在此定义的一个属性，用于将一个持有者指派给一个角色。有关角色属性值的规范不在本规范的讨论范围之内。

```
role ATTRIBUTE  ::= {
    WITH SYNTAX             RoleSyntax
    ID                      id-at-role }

RoleSyntax ::=      SEQUENCE {
roleAuthority       [0]     GeneralNames    OPTIONAL,
roleName            [1]     GeneralName }
```

本特权属性将用于组装角色指派证书的 **attributes** 字段。如果角色指派证书为公开密钥证书，那么本属性将用于组装该公开密钥证书的 **subjectDirectoryAttributes** 扩展。

如果出现，那么 **roleAuthority** 确定已得到认可的机构，它负责发放角色规范证书。

如果 **roleAuthority** 出现，并且特权验证者使用一个角色规范证书来确定指派给角色的特权，那么在该角色规范证书的 **issuer** 字段中应至少出现一个在 **roleAuthority** 中的名称。如果特权验证者使用角色规范证书之外的方法来确定指派给角色的特权，那么确保由在本部件中命名的机构来指派这些特权的机制将不在本规范的讨论范围之内。

如果 **roleAuthority** 不出现，那么将通过其他方法来确定负责的机构的身份。在角色规范证书用于向角色指派特权的情况下，角色指派证书中的 **roleSpecCertIdentifier** 扩展是一种实现该绑定的方法。

roleName 部件用于确定包含该属性的角色指派证书的持有者向哪个角色指派特权。如果特权验证者使用一个角色规范证书来确定指派给该角色的特权，那么该角色名称还将出现在角色规范证书的 **holder** 字段中。

14.5 XML特权信息属性

特权规范通常是一个应用特定的问题，它不在本规范的讨论范围之内。本属性不定义任何特殊的特权信息，它提供了一个容器属性，在其中可以在属性证书中传达 XML 编码的特权。

```
xmlPrivilegeInfo    ATTRIBUTE  ::= {
  WITH SYNTAX       UTF8String -- 包含 XML 编码的特权信息
  ID                id-at-xMLPrivilegeInfo }
```

可以利用 ASN.1 或 XSD 来定义角色属性类型的 XML 方案。

包含在 **UTF8String** 中的 XML 需要是自鉴别的。

以下是一个 ASN.1 方案，用于定义一个 XML 角色属性类型。后跟一个有关相同属性类型的 XSD 规范，以及一个 XML 实例例子。实例例子是一个对 ASN.1 和 XSD 方案实例都是有效的实例，可以通过 ASN.1 或 XSD 工具进行证实。

例中的方案定义了一个带 ID、发布机构和角色名称的角色属性。

```
CERTIFICATE-ATTRIBUTE DEFINITIONS ::=
BEGIN
  Role ::= [UNCAPITALIZED] SEQUENCE {
      id              [ATTRIBUTE] XML-ID,
      authorities       SEQUENCE (1..MAX) OF
                        authority UTF8String,
      name              UTF8String }

    XML-ID  ::= UTF8String
END
```

以下 XSD 方案是一个可选的（严格等同）定义：

```
<schema xmlns="http://www.w3.org/2000/08/XMLSchema">
 <element name="role">
  <attribute name="id" type="ID"/>
  <complexType>
   <sequence>
    <element name="authorities">
     <complexType>
      <sequence>
       <element name="authority" type="string" minOccurs="1" maxOccurs="*"/>
      </sequence>
     </complexType>
    </element>
    <element name="name" type="string"/>
   </sequence>
  </complexType>
 </element>
</schema>
```

与上述方案定义一致的一个实例的例子如下所示，它将是 **xMLPrivilegeInfo** 属性类型的一个值：

```
<role id="123" xmlns="http://www.example.org/certificates/attribute">
<authorities>
<authority>Fictitious Organization</authority>
</authorities>
<name>manager</name>
</role>
```

15 特权管理证书扩展

出于特权管理之目的，可以在证书中包括以下证书扩展。与扩展本身定义一起，还提供了有关证书类型（在其中可以出现扩展）的规则。

除了 SOA 标识符扩展，对任何可以包括在公开密钥证书中的扩展，都只有在公开密钥证书为向其对象指派特权的公开密钥证书时（即出现 **subjectDirectoryAttributes** 扩展），才能包括在内。如果在公开密钥证书中出现任何这些扩展，那么该扩展适用于出现在 **subjectDirectoryAttributes** 扩展中的所有特权。

用于公布属性证书撤消通知的撤消清单（ACRL 和 AARL）可以包含任何 CRL 或 CRL 条目扩展，定义这些条目是为了供本规范第 2 部分中的 CRL 和 CARL 使用。

本节用于规定以下领域中的扩展：

a） 基本的特权管理：这些证书扩展用于传达与特权声明有关的信息。

b） 特权撤消：这些证书扩展用于传达有关确定撤消状态信息的信息。

c） 机构源：这些证书扩展与特权指派的信任源有关，针对的是验证者对某个特定资源的特权指派。

d） 角色：这些证书扩展用于传达有关确定相关角色规范证书的信息。

e） 委托：这些证书扩展允许对所指派特权的后续委托设置约束。

15.1 基本的特权管理扩展

15.1.1 需求

以下需求与基本的特权管理有关：

a） 发放者需要能够在可以声明特权期间对时间设置约束；

b） 发放者需要能够将属性证书面向特定的服务器/服务；

c） 可能需要发放者将计划用于显示的信息传送给使用证书的特权声明者与/或特权验证者；

d） 发放者需要能够对特权策略设置约束，利用这些特权策略就可以使用所指派的特权。

15.1.2 基本的特权管理扩展字段

定义了以下扩展字段：

a） 时间规范；

b） 目标信息；

c） 用户通知；

d） 可接受的特权策略；

e） 间接的发布者；

f） 无声明。

15.1.2.1 时间规范扩展

一个 AA 可以使用时间规范扩展来限制特定的时间周期，在该时间周期内，特权持有者可以声明在包含本扩展的证书中指派的特权。例如，一个 AA 可以发放一个用于指派特权的证书，这些特权只能在星期一与星期五之间以及上午 9 点与下午 5 点之间声明。再如，在委托情况下，可以是一个管理者，在他离开休假期间，将签署机构委托给一个下属机构。

本字段定义如下：

```
timeSpecification EXTENSION  ::= {
    SYNTAX              TimeSpecification
    IDENTIFIED BY       id-ce-timeSpecification }
```

本扩展可以出现在由 AA（包括 SOA）发放给各实体的属性证书或公开密钥证书中，这些实体可以作为特权声明者，包括其他 AA 和终端实体。本扩展不得包括在包含 SOA 标识符扩展的证书中或发放给 AA（它们不可以作为特权声明者）的证书中。

如果本扩展出现在发放给某个实体（作为一个 AA）的证书中，那么它只适用于包含在证书中的该实体特权声明。它对以下时间周期没有影响，即在该时间周期内 AA 能够发放证书。

由于本扩展能够有效规定包含它的证书的有效周期，因此本扩展将被标记为关键的（即通过包括本扩展，发放者可以明确定义在规定时间之外的特权指派是无效的）。

如果本扩展出现，但不被特权验证者所理解，那么证书将被拒绝。

15.1.2.1.1 时间规范匹配

时间规范匹配规则比较出现的值是否与类型 **AttributeCertificate** 的属性值相同。

```
timeSpecificationMatch MATCHING-RULE  ::= {
    SYNTAX          TimeSpecification
    ID              id-mr-timeSpecMatch }
```

如果保存的值包含 **timeSpecification** 扩展，并且如果出现在出现值中的部件匹配于保存值中的对应部件，那么本匹配规则将返回 TRUE。

15.1.2.2 目标信息扩展

目标信息扩展能够使属性证书面向一个特定的服务器/服务集。包含本扩展的属性证书应该只能在规定的服务器/服务上可用。

本字段定义如下：

```
targetingInformation  EXTENSION  ::= {
    SYNTAX              SEQUENCE SIZE (1..MAX) OF Targets
    IDENTIFIED BY       id-ce-targetInformation }

Targets     ::=       SEQUENCE SIZE (1..MAX) OF Target

Target      :      := CHOICE {
    targetName          [0]     GeneralName,
    targetGroup         [1]     GeneralName,
    targetCert          [2]     TargetCert }

TargetCert  ::=    SEQUENCE {
    targetCertificate   IssuerSerial,
    targetName          GeneralName OPTIONAL,
    certDigestInfo      ObjectDigestInfo OPTIONAL }
```

如果出现，那么 **targetName** 部件用于提供目标服务器/服务的名称，包含属性证书面向的就是这些服务器/服务。

如果出现，那么 **targetGroup** 部件用于提供目标组的名称，包含属性证书面向的就是这些目标组。如何确定 **targetGroup** 内目标的成员资格不在本规范的讨论范围之内。

如果出现，那么 **targetCert** 部件用于确定提及其证书的目标服务器/服务。

本扩展可以出现在由 AA（包括 SOA）发放给各实体的属性证书中，这些实体可以作为特权声明者，包括其他 AA 和终端实体。本扩展不得包括在公开密钥证书中或发放给 AA（它们不可以作为特权声明者）的属性证书中。

如果本扩展出现在发放给某个实体（作为一个 AA）的属性证书中，那么它只适用于包含在证书中的该实体特权声明。它对 AA 发放证书的能力没有影响。

本扩展总是为关键的。

如果本扩展出现，但特权验证者不在规定的特权验证者之中，那么应该拒绝属性证书。

如果本扩展不出现，那么不面向属性证书，并可以被任何服务器所接受。

15.1.2.3 用户通知扩展

用户通知扩展使一个 AA 能够包括一个通知，当声明其特权时，该通知应该显示给持有者，与/或当使用包含本扩展的属性证书时，该通知应该显示给特权验证者。

本字段定义如下：

```
userNotice  EXTENSION  ::= {
    SYNTAX              SEQUENCE SIZE (1..MAX) OF UserNotice
    IDENTIFIED BY       id-ce-userNotice }
```

本扩展可以出现在由 AA（包括 SOA）发放给各实体的属性证书或公开密钥证书中，这些实体可以作为特权声明者，包括其他 AA 和终端实体。本扩展不得包括在包含 SOA 标识符扩展的证书中或发放给 AA（它们不可以作为特权声明者）的证书中。

如果本扩展出现在发放给某个实体（作为一个 AA）的证书中，那么它只适用于包含在证书中的该实体特权声明。它对 AA 发放证书的能力没有影响。

由证书发放者选择决定，本扩展可以是关键的或者是非关键的。

如果本扩展标志为关键的，那么用户通知将在每次声明特权时显示给特权验证者。如果特权声明者向特权验证者提供属性证书（即特权验证者不直接从某个知识库中对它进行检索），那么用户通知还将显示给特权声明者。

如果本扩展标志为非关键的，那么在证书中声明的特权可以由特权验证者授予，而不管用户通知是否显示给特权声明者与/或特权验证者。

15.1.2.4 可接受的特权策略扩展

可接受的特权策略字段用于约束所指派特权的声明，以供某个特定的特权策略集使用。

本字段定义如下：

```
acceptablePrivilegePolicies EXTENSION ::= {
    SYNTAX              AcceptablePrivilegePoliciesSyntax
    IDENTIFIED BY       id-ce-acceptablePrivilegePolicies }

AcceptablePrivilegePoliciesSyntax       ::=     SEQUENCE SIZE (1..MAX) OF PrivilegePolicy
```

本扩展可以出现在由 AA（包括 SOA）发放给其他 AA 或终端实体的属性证书或公开密钥证书中。如果本扩展包括在一个公开密钥证书中，那么它仅与对象的以下能力有关，即作为 **subjectDirectoryAttributes** 扩展中所含特权的特权声明者。

如果出现，那么本扩展将被标志为关键的。

如果本扩展出现，并被特权验证者所理解，那么验证者将确保其特权正在接受比较的特权策略为本扩展中所确定的特权策略之一。

如果本扩展出现，但不被特权验证者所理解，那么证书将被拒绝。

15.1.2.5 间接的发放者扩展

在某些环境中，可以间接地委托特权。在这些情况下，委托者请求 AA 发放一个代表它们向另一个实体委托特权的证书。间接发放者字段用在属性证书中或由 SOA 发放给 AA 的公开密钥证书中。出现本扩展意味着该 SOA 授权对象 AA 充当一个代理，并代表其他委托者来发放用于委托特权的证书。

```
indirectIssuer EXTENSION ::= {
    SYNTAX          BOOLEAN
    IDENTIFIED BY   id-ce-indirectIssuer }
```

本扩展总是为非关键的。

间接发放者匹配规则比较出现的值是否与类型 **AttributeCertificate** 的属性值相同。

```
indirectIssuerMatch MATCHING-RULE  ::= {
SYNTAX      BOOLEAN
ID          id-mr-indirectIssuerMatch }
```

如果保存的值包含 **indirectIssuer** 扩展，并且如果出现在出现值中的值匹配于保存的值，那么本匹配规则将返回 TRUE。

15.1.2.6 无声明扩展

如果出现，那么本扩展指明，AC 持有者不能声明在 AC 属性中指明的各特权。本字段只能插入 AA ACs 中，不能插入终端实体 ACs 中。如果出现，本扩展将总是标记为关键的。

```
noAssertion EXTENSION ::= {
SYNTAX NULL
IDENTIFIED BY  id-ce-noAssertion }
```

15.2 特权撤消扩展

15.2.1 需求

以下需求与属性证书的撤消有关：

a） 为了控制 CRL 规模，可能需要指派所有证书集的子集，各证书由一个 AA 发放给不同的 CRL；

b） 属性证书发放者需要能够在属性证书中指明，没有任何撤消信息可用于该证书。

15.2.2 特权撤消扩展字段

定义了以下扩展字段：

a） *CRL 分发点*；

b） *无撤消信息*。

15.2.2.1 CRL分发点扩展

CRL 分发点扩展在本规范的第 2 部分中定义，用在公开密钥证书中。本字段还可以包括在一个实现证书中。它可以出现在发放给 AA（包括 SOA）的证书中，以及发放给终端实体的证书中。

如果出现在一个证书中，那么特权验证者将以第 2 部分中所述的、与公开密钥证书完全相同的方式来处理本扩展。

15.2.2.2 无撤消信息扩展

在某些环境中（例如，当以非常短的有效期来发放属性证书时），可能无需撤消证书。一个 AA 可以使用本扩展来指明撤消状态信息不是为本属性证书提供。本字段定义如下：

```
noRevAvail EXTENSION ::= {
    SYNTAX              NULL
    IDENTIFIED BY       id-ce-noRevAvail }
```

本扩展可以出现在由 AA（包括 SOA）发放给各终端实体的属性证书中。本扩展不得包括在发放给 AA 的公开密钥证书或属性证书中。

本扩展总是为非关键的。

如果本扩展出现在一个属性证书中，那么特权验证者无需寻找撤消状态信息。

15.3 机构源扩展

15.3.1 需求

以下需求与机构源有关：

a） 在某些环境中，CA 需要严格控制能作为 SOA 的实体；

b） 需要做出有效的、有关特权属性的语法定义和控制规则，它们对负责的 SOA 是可用的。

15.3.2 SOA扩展字段

定义了以下扩展字段：

a） *SOA* 标识符；

b） 属性描述符。

15.3.2.1 SOA标识符扩展

SOA 标识符扩展指明，出于特权管理之目的，证书对象可以作为一个 SOA。这样，证书对象就可以定义各属性，用于指派特权，发放有关这些属性的属性标识符证书，以及使用对应已认证公开密钥的专用密钥来发放用于向持有者指派特权的证书。这些后续的证书可以是属性证书或者是公开密钥证书，带一个包含特权的 **subjectDirectoryAttributes** 扩展。

在某些环境中，不需要本扩展，可以使用其他机制来确定可以作为 SOA 的实体。只在以下环境中需要本扩展，即要求 CA 实施严格的集中式控制来管理作为 SOA 的实体。

本字段定义如下：

```
sOAIdentifier EXTENSION  ::= {
    SYNTAX              NULL
    IDENTIFIED BY       id-ce-sOAIdentifier }
```

如果本扩展未出现在证书中，那么将用其他方式来确定对象/持有者作为一个 SOA 的能力。

本字段只能出现在一个发放给 SOA 的公开密钥证书中。它不得包括在发放给其他 AA 或终端实体特权持有者的属性证书或公开密钥证书中。

交叉认证仅适用于公开密钥证书，并适用于属性证书。因此，发放给 CA（为包含 SOA 标识符扩展之证书的发放者）的交叉证书不向在本扩展中确定的 SOA 提供过渡信任。

本扩展总是为非关键的。

15.3.2.1.1 SOA标识符匹配

匹配规则的 SOA 标识符对出现值和类型 **Certificate** 的属性值进行比较，看它们是否相同。

```
sOAIdentifierMatch MATCHING-RULE ::= {
  SYNTAX      NULL
  ID          id-mr-sOAIdentifierMatch }
```

如果保存的值包含一个 SOA 标识符扩展，那么本匹配规则返回 TRUE。

15.3.2.2 属性描述符扩展

特权验证者需要定义特权属性以及用于管理该特权后续委托的控制规则，以确保正确的授权。这些定义和规则可以用一系列本规范之外的方法来提供给特权验证者（例如，可以在特权验证者上来本地配置它们）。

本扩展提供了一种机制，SOA 可以用它来定义特权属性以及相关的、特权验证者可用的控制规则。包含本扩展的属性证书称为属性描述符证书，是一种特殊类型的属性证书。虽然语法上等同于一个 **AttributeCertificate**，但一个属性描述符证书应：

— 在其 **attributes** 字段中包含一个空的 **SEQUENCE**；

— 是一个自发放的证书（即发放者和持有者为同一实体）；以及

— 包括属性描述符扩展。

本字段定义如下：

```
attributeDescriptor EXTENSION  ::= {
    SYNTAX              AttributeDescriptorSyntax
    IDENTIFIED BY       {id-ce-attributeDescriptor } }
```

```
AttributeDescriptorSyntax  ::= SEQUENCE {
   identifier                  AttributeIdentifier,
   attributeSyntax             OCTET STRING (SIZE(1..MAX)),
   name                [0]     AttributeName   OPTIONAL,
   description         [1]     AttributeDescription    OPTIONAL,
   dominationRule              PrivilegePolicyIdentifier}

AttributeIdentifier  ::= ATTRIBUTE.&id({AttributeIDs})

AttributeIDs ATTRIBUTE  ::= {...}

AttributeName  ::= UTF8String(SIZE(1..MAX))

AttributeDescription  ::= UTF8String(SIZE(1..MAX))

PrivilegePolicyIdentifier      ::=    SEQUENCE {
    privilegePolicy                      PrivilegePolicy,
    privPolSyntax                        InfoSyntax }
```

attributeDescriptor 扩展值的 **identifier** 部件是用于确定属性类型的对象标识符。

attributeSyntax 部件包含属性句法的 ASN.1 定义。提供这样一个 ASN.1 定义用于规定匹配规则可操作属性的信息部件，在 ITU-T X.501 建议书 | ISO/IEC 9594-2。

name 部件可选地包含一个用户友好的名称，通过它属性可以得到认可。

description 部件可选地包含一个用户友好的属性描述。

dominationRule 部件规定，对属性，它对委托的特权意味着“小于”委托者所持有的对应特权。**privilegePolicy** 部件通过其对象标识符来确定包含规则的特权策略的实例。**privPolSyntax** 部件包含特权策略自身，或者包含一个指向某个位置的指针，在该位置能找到此部件。如果包括一个指针，那么还可以包括一个特权策略的可选散列，以便允许对所参考的特权策略进行完整性检查。

本扩展只可以出现在属性标识符证书中。本扩展不得出现在 SOA 自发放证书之外的公开密钥证书或属性证书中。

本扩展总是为非关键的。

由 SOA 在创建/定义对应属性类型时创建的属性描述符证书是一种方式，利用它，可以理解“向下”委托的一般性约束，并在基础设施中执行这些约束。在号码簿中，包含本扩展的属性证书将保存在 SOA 号码簿条目的 **attributeDescriptorCertificate** 属性中。

15.3.2.2.1 属性描述符匹配

匹配规则的属性描述符对出现值和类型 **AttributeCertificate** 的属性值进行比较，看它们是否相同。

```
attDescriptor MATCHING-RULE  ::= {
    SYNTAX          AttributeDescriptorSyntax
    ID              id-mr-attDescriptorMatch }
```

如果保存的值包含 **attributeDescriptor** 扩展，并且如果出现在出现值中的部件匹配于保存值中的对应部件，那么本匹配规则将返回 TRUE。

15.4 角色扩展

15.4.1 需求

以下需求与角色有关：

— 如果证书是一个角色指派证书，那么特权验证者需要能够确定对应的角色规范证书，它包含指派给角色本身的特定特权。

15.4.2 角色扩展字段

定义了以下扩展字段：

— 角色规范证书标识符。

15.4.2.1 角色规范证书标识符扩展

AA 可以将本扩展作为一个指针使用，指向一个角色规范证书，它包含对某个角色的特权指派。它可以出现在一个角色指派证书中（即一个包含 **role** 属性的证书）。

当处理一个角色指派证书时，一个特权验证者需要获得该角色的特权集，以便确定是否通过验证。如果特权指派给一个角色规范证书中的角色，那么可以使用本字段来确定该证书。

本字段定义如下：

```
roleSpecCertIdentifier EXTENSION ::=
    {
    SYNTAX              RoleSpecCertIdentifierSyntax
    IDENTIFIED BY       { id-ce-roleSpecCertIdentifier }
}

RoleSpecCertIdentifierSyntax  ::= SEQUENCE SIZE (1..MAX) OF RoleSpecCertIdentifier

RoleSpecCertIdentifier  ::= SEQUENCE {
    roleName                    [0]     GeneralName,
    roleCertIssuer              [1]     GeneralName,
    roleCertSerialNumber        [2]     CertificateSerialNumber     OPTIONAL,
    roleCertLocator             [3]     GeneralNames                OPTIONAL
    }
```

roleName 用于确定角色。该名称将与本扩展所参考的角色规范证书 **holder** 部件中的名称相同。

roleCertIssuer 用于确定发放参考角色规范证书的 AA。

如果出现，那么 **roleCertSerialNumber** 包含角色规范证书的序列号。注意：如果指派给角色自身的特权发生了变化，那么将向角色发放一个新的角色规范证书。而后将需要用参考了新的序列号的证书来替换包含本扩展的任何证书，包括 **roleCertSerialNumber** 部件。虽然在某些环境中需要该行为，但在许多其他环境中并不要求这么做。典型地，本部件将不出现，使得能够自动更新指派给角色自身的特权，而不对角色指派证书造成影响。

如果出现，那么 **roleCertLocator** 包含可用于确定角色规范证书的信息。

本扩展可以出现在角色指派证书中，这些证书可以是由 AA（包括 SOA）发放给其他 AA 或终端实体特权持有者的属性证书或公开密钥证书。本扩展不得包括在包含 SOA 标识符扩展的证书中。

如果出现，那么特权验证者可以使用本扩展来确定角色规范证书。

如果本扩展未出现，那么：

a） 将使用其他方式来确定角色规范证书；或者

b） 使用角色规范证书之外的机制来为角色指派特权（例如，可以在特权验证者上本地配置角色特权）。

本扩展总是为非关键的。

15.4.2.1.1 角色规范证书ID匹配

匹配规则的角色规范证书标识符对出现值和类型 **AttributeCertificate** 的属性值进行比较，看它们是否相同。

```
roleSpecCertIdMatch MATCHING-RULE  ::= {
    SYNTAX          RoleSpecCertIdentifierSyntax
    ID              id-mr-roleSpecCertIdMatch }
```

如果保存的值包含 **roleSpecCertIdentifier** 扩展，并且如果出现在出现值中的部件匹配于保存值中的对应部件，那么本匹配规则将返回 TRUE。

15.5 委托扩展

15.5.1 需求

以下需求与特权委托有关：

a） 终端实体特权证书需要能够区别于 AA 证书，以防止在未经授权的情况下终端实体将其自身建为 AA。还需使 AA 有可能限制后续委托通路的长度；

b） AA 需要能够规定适当的名称空间，使在其中能够实现特权委托。对这些约束是否得到了遵守，特权验证者需要能够进行检查；

c） AA 需要能够规定可接受的证书策略，当该 AA 声明一个特权委托时，进一步下放委托通路的特权声明者将用之来对其自身进行鉴权；

d） 特权验证者需要能够确定对应的属性证书，以便发放者确保发放者拥有足够的特权来委托当前证书中的特权；

e） 需要一个独立的委托服务（DS）来发放委托特权的证书，同时 DS 服务器自身不能作为这些特权的要求者。

15.5.2 委托扩展字段

定义了以下扩展字段：

a） 基本的属性约束；

b） 委托的名称约束；

c） 可接受的证书策略；

d） 机构的属性标识符；

e） 间接的发布者；

f） 代表某某发布的。

15.5.2.1 基本的属性约束扩展

本字段用于指明是否允许后续的特权委托，这些特权在包含本扩展的证书中予以指派。如果允许，那么还可以对委托通路长度约束做出规定。

本字段定义如下：

```
basicAttConstraints EXTENSION ::=
{
      SYNTAX                BasicAttConstraintsSyntax
      IDENTIFIED BY         { id-ce-basicAttConstraints }
}

BasicAttConstraintsSyntax ::= SEQUENCE
   {
      authority             BOOLEAN DEFAULT FALSE,
      pathLenConstraint     INTEGER (0..MAX) OPTIONAL
   }
```

authority 部件用于指明持有者是否有权进一步委托特权。如果 **authority** 为 **TRUE**，那么持有者也是一个 AA，并有权依据相关的约束进一步委托特权。如果 **authority** 为 **FALSE**，那么持有者是一个终端实体，并无权委托特权。

只有当 **authority** 设置为 **TRUE** 时，**pathLenConstraint** 部件才有意义。它给出了在委托通路中可以跟随本证书的最大 AA 证书数量。值 0 指明该证书的对象只可以将证书发放给终端实体而可以发放给 AA。如果在委托通路的任何证书中都未出现 **pathLenConstraint** 字段，那么对委托通路允许的长度没有限制。注意：约束从通路中的下一个证书开始发挥作用。约束用于控制包含约束的 AA 证书与终端实体证书之间的 AA 证书数量。约束用于限制包含本扩展的证书与终端实体证书之间的委托通路片段长度。它对信任锚点与包含本扩展的证书之间委托通路中的 AA 证书数量没有任何影响。因此，完整委托通路的长度可以超过由本扩展约束的最大片段长度。约束用于控制包含约束的 AA 证书与终端实体证书之间的 AA 证书数量。因此，通路本片段总的长度可以超过约束值 2 个证书。（这包括片段两个端点上的证书加上两个端点之间的 AA 证书，它们受本扩展值的约束。）

本扩展可以出现在由 AA（包括 SOA）发放给其他 AA 或终端实体的属性证书或公开密钥证书中。本扩展不得包括在包含 SOA 标识符扩展的证书中。

如果本扩展出现在一个属性证书中，并且 **authority** 为 **TRUE**，那么持有者有权发放后续属性证书（它们可以将包含的特权发放给其他实体），但不能发放公开密钥证书。

如果本扩展出现在一个公开密钥证书中，并且如果 **basicConstraints** 扩展指明对象也是一个 CA，那么对象有权发放后续公开密钥证书（它们将这些特权委托给其他实体），但不能发放属性证书。如果包括一个通路长度约束，那么对象只可以在本扩展规定的约束交集以及 **basicConstraints** 扩展中规定的任何范围内进行委托。如果本扩展出现在一个公开密钥证书中，但 **basicConstraints** 扩展不出现，或者指明对象是一个终端实体，那么对象无权委托特权。

本扩展可以由证书发放者选择决定是关键的还是非关键的。建议将之标志为关键的，否则未授权作为 AA 的持有者可以发放证书，并且特权验证者可以无意地使用这样一个证书。

如果本扩展出现并被标志为关键的，那么：

— 如果 **authority** 的值未被设置为 **TRUE**，那么委托的属性不得用于进一步委托；

— 如果 **authority** 的值被设置为 **TRUE**，并且 **pathLenConstraint** 出现，那么特权验证者将检查正在处理的委托通路与 **pathLenConstraint** 的值是否一致。

如果本扩展出现，并标志为非关键的，并且不被特权验证者认可，那么该系统应使用其他方式来确定委托的属性是否可用于更进一步的委托。

如果本扩展未出现，或者如果出现的扩展带一个空的 **SEQUENCE** 值，那么规定持有者只能是一个终端实体而不能是一个属性机构，并且持有者不允许对包含在属性证书中的特权进行任何委托。

15.5.2.1.1 基本的属性约束匹配

匹配规则的基本属性约束对出现值和类型 **AttributeCertificate** 的属性值进行比较，看它们是否相同。

```
basicAttConstraintsMatch MATCHING-RULE ::= {
    SYNTAX          BasicAttConstraintsSyntax
    ID              id-mr-basicAttConstraintsMatch }
```

如果保存的值包含 **basicAttConstraints** 扩展，并且如果出现在出现值中的部件匹配于保存值中的对应部件，那么本匹配规则将返回 TRUE。

15.5.2.2 委托的名称约束扩展

委托的名称约束字段用于指明一个名称空间，在该名称空间内，需要确定委托通路中各后续证书中所有持有者的名称。

本字段定义如下：

```
delegatedNameConstraints EXTENSION ::= {
    SYNTAX              NameConstraintsSyntax
    IDENTIFIED BY       id-ce-delegatedNameConstraints }
```

本扩展以与公开密钥证书中 **nameConstraints** 扩展相同的方式进行处理。如果 **permittedSubtrees** 出现，那么对委托通路中持有者 AA 和各后续 AA 发放的所有属性证书，只有那些持有者名称在这些子树中的属性证书才是可接受的。如果 **excludedSubtrees** 出现，那么委托通路中持有者 AA 或各后续 AA 发放的、在这些子树中拥有一个持有者名称的任何属性证书都是不可接受的。如果 **permittedSubtrees** 和 **excludedSubtrees** 都出现，并且名称空间重叠，那么首先需要发表排除在外声明。

本扩展可以出现在由 AA（包括 SOA）发放给其他 AA 的属性证书或公开密钥证书中。本扩展不得包括在发放给终端实体的证书或包含 SOA 标识符扩展的证书中。

如果本扩展出现在公开密钥证书中，并且如果 **nameConstraints** 扩展也出现，那么对象只可以在本扩展规定的以及 **nameConstraints** 扩展规定的约束交集内进行委托。

本扩展可以由属性证书发放者选择决定是关键的还是非关键的。建议将之标志为关键的，否则属性证书用户无法检查委托通路中的各后续属性证书是否位于发放 AA 计划的名称空间中。

15.5.2.2.1 委托的名称约束匹配

匹配规则的委托名称约束对出现值和类型 **AttributeCertificate** 的属性值进行比较，看它们是否相同。

```
delegatedNameConstraintsMatch MATCHING-RULE  ::= {
    SYNTAX          NameConstraintsSyntax
    ID              id-mr-delegatedNameConstraintsMatch}
```

如果保存的值包含 **attributeNameConstraints** 扩展，并且如果出现在出现值中的部件匹配于保存值中的对应部件，那么本匹配规则将返回 TRUE。

15.5.2.3 可接受的证书策略扩展

在带属性证书的委托中，可接受的证书策略用于控制可接受的证书策略，依据这些策略，需要发放委托通路中后续持有者的公开密钥证书。通过在本字段中列举一系列策略，AA 要求委托通路中的各后续发放者只将所包含的特权委托给拥有公开密钥证书（它们依据一个或多个所列举的证书策略发放）的各持有者。

本字段定义如下：

```
acceptableCertPolicies EXTENSION ::= {
    SYNTAX              AcceptableCertPoliciesSyntax
    IDENTIFIED BY       id-ce-acceptableCertPolicies }

AcceptableCertPoliciesSyntax ::= SEQUENCE SIZE (1..MAX) OF CertPolicyId

CertPolicyId ::= OBJECT IDENTIFIER
```

本扩展只能出现在由 AA（包括 SOA）发放给其他 AA 的属性证书中。本扩展不得包括在终端实体属性证书或任何公开密钥证书中。在使用公开密钥证书的情况下，该相同功能由 **certificatePolicies** 和其他相关的扩展提供。

如果出现，那么本扩展将被标志为关键的。

如果本扩展出现，并为特权验证者所理解，那么验证者将确保依据一个或多个列举的证书策略、利用一个公开密钥证书，对委托通路中的所有后续特权声明者进行鉴权。

如果本扩展出现，但不被特权验证者所理解，那么证书将被拒绝。

15.5.2.3.1 可接受的证书策略匹配

匹配规则的可接受证书策略对出现值和类型 **AttributeCertificate** 的属性值进行比较，看它们是否相同。

```
acceptableCertPoliciesMatch MATCHING-RULE  ::= {
    SYNTAX          AcceptableCertPoliciesSyntax
    ID              id-mr-acceptableCertPoliciesMatch }
```

如果保存的值包含 **acceptableCertPolicies** 扩展，并且如果出现在出现值中的部件匹配于保存值中的对应部件，那么本匹配规则将返回 TRUE。

15.5.2.4 机构属性标识符扩展

在特权委托中，委托特权的 AA 自身至少需拥有相同的、用于委托该特权的特权和授权。向另一个 AA 或一个终端实体委托特权的 AA 可以将本扩展置于它发放的 AA 或终端实体证书中。本扩展是一个指向证书的向后指针，在其中，为包含扩展的证书发放者指派其对应的特权。特权验证者可使用本扩展来确保发放 AA 有足够的特权来向包含本扩展的证书持有者做委托。

本字段定义如下：

```
authorityAttributeIdentifier EXTENSION  ::=
    {
    SYNTAX                AuthorityAttributeIdentifierSyntax
    IDENTIFIED BY         { id-ce-authorityAttributeIdentifier }
    }

AuthorityAttributeIdentifierSyntax    ::=        SEQUENCE SIZE (1..MAX) OF AuthAttId

AuthAttId         ::=        IssuerSerial
```

包含本扩展的证书可以包括对证书持有者的多个特权委托。如果在多个证书中对发放本证书的 AA 指派这些特权，那么本扩展将包括多个指针。

本扩展可以出现在由 AA 发放给其他 AA 或终端实体特权持有者的属性证书或公开密钥证书中。本扩展不得包括在由 SOA 发放的证书中或者包含 SOA 标识符扩展的公开密钥证书中。

本扩展总是为非关键的。

15.5.2.4.1 AA标识符匹配

匹配规则的机构属性标识符对出现值和类型 **AttributeCertificate** 的属性值进行比较，看它们是否相同。

```
authAttIdMatch MATCHING-RULE  ::= {
    SYNTAX            AuthorityAttirbuteIdentifierSyntax
    ID                id-mr-authAttIdMatch }
```

如果保存的值包含 **authorityAttributeIdentifier** 扩展，并且如果出现在出现值中的部件匹配于保存值中的对应部件，那么本匹配规则将返回 TRUE。

15.5.2.5 间接的发放者扩展

在某些环境中，可以间接地委托特权。在这些情况下，委托者请求 DS 服务器发放一个代表它们向另一个实体委托特权的证书。间接发放者字段用在属性证书中或由 SOA 发放给 DS 服务器的公开密钥证书中。出现本扩展意味着该 SOA 授权对象 AA（DS 服务器）充当一个代理，并代表其他委托者来发放用于委托特权的证书。

```
indirectIssuer EXTENSION ::= {
    SYNTAX            NULL
    IDENTIFIED BY         id-ce-indirectIssuer }
```

本扩展总是为非关键的。

匹配规则的间接发放者对出现值和类型 **AttributeCertificate** 的属性值进行比较，看它们是否相同。

```
indirectIssuerMatch MATCHING-RULE  ::= {
    SYNTAX            NULL
    ID          id-mr-indirectIssuerMatch }
```

如果保存的值包含 **indirectIssuer** 扩展，并且如果出现在出现值中的值匹配于保存值，那么本匹配规则将返回 TRUE。

15.5.2.6 代表某某发布的

本扩展由间接发放者（DS 服务器）插入到某个 AC 中。它表明 AA 已请求 DS 服务器发放 AC，并允许构造和验证委托链。

```
issuedOnBehalfOf EXTENSION ::= {
    SYNTAX GeneralName
    ID id-ce-issuedOnBehalfOf }
```

`GeneralName` 为请求间接发放者（DS 服务器）发放本 AC 的 AA 的名称。

本 AC 的发放者必须已通过其 AC 中的 IndirectIssuer 扩展，授权由 SOA 代表其他 AA 来发放 ACs。

本扩展可以是关键的或非关键的，根据需要来确保委托通路验证。

16 特权通路处理程序

特权通路处理由特权验证者负责完成。属性证书的通路处理规则在某种程度上类似于公开密钥证书的通路处理规则。

未在本节中论述的其他通路处理部件包括证书签名验证、证书有效期确认等。

对由一个单个证书组成的特权通路（即特权由 SOA 直接指派给特权声明者），只需下面第 16.1 节中所述的基本程序，除非特权指派给某个角色。在这种情况下，如果特权验证者未以角色的特定特权进行配置，那么它需要获得角色规范证书，将特定特权指派给角色，如下面第 16.2 节中所述。如果由一个中间 AA 向特权声明者委派其特权，那么还需要第 16.3 节中的委托通路程序。这些程序不是连续执行的。在确定所声明的特权对基本程序内使用范畴是否充分之前执行角色处理程序和委托处理程序。

16.1 基本的处理程序

将对通路中每个证书的签名进行验证。在本节中不重复与签名验证和公开密钥证书相关的程序。特权验证者将使用第 10 节中的程序对通路中每个实体的身份进行验证。注意：对属性证书中的签名进行检查必然涉及对参考之公开密钥证书的有效性进行检查。当使用属性证书指派特权时，在确定特权声明者属性证书的最终有效性期间，通路处理引擎将需要考虑 PMI 和 PKI 的元素。一旦确认有效性，那么就可以依据相关特权策略与其他证书使用范畴相关信息的比较结果，使用该证书中所含的特权。

使用范畴将决定特权持有者是否真的想声明所含的特权，以供该范畴使用。对信任 SOA 存在一条证书链的事实本身不足以做出该决定。特权持有者使用该证书的愿望必须明确指明并加以验证。不过，确保特权声明者已准确证实这样一个特权声明的机制已超出本规范的范围。作为一个例子，如果特权持有者签署一个对该证书的参考，以指明其对该范畴使用该证书的愿望，那么对这样一个特权声明是可以验证的。

对通路中每个不包含 **noRevAvail** 扩展的属性证书，特权声明者将确保属性证书未被撤消。

特权声明者将确保所声明的特权对称为“评估时间”的时间而言是有效的，“评估时间”可以是任何时间，即当前的检查时间或过去的任何时间。对访问控制服务而言，总对当前时间进行检查。不过，对不可否认而言，可以对过去的某个时间或当前时间进行检查。当证书得到验证时，特权验证者将确保评估时间落在通路中使用的所有证书的所有有效周期内。另外，如果通路中的任何证书都包含 **timeSpecification** 扩展，那么有关特权可声明次数的约束也需使得特权声明在评估之时是有效的。

如果 **targetingInformation** 扩展出现在用于声明某个特权的证书中，那么特权验证者将检查它正在验证的服务器/服务是否包括在目标清单中。

如果证书是一个角色指派证书，那么第 16.2 节中所述的处理程序需确保确定适当的特权。如果特权委派给实体，而不是直接由特权验证者所信任的 SOA 来指派，那么第 16.3 节中所述的处理程序需确保正确完成委派。

特权验证者还将确定所声明的特权对使用范畴而言是否足够。特权策略负责建立有关做出该决定的规则，并包括有关需要考虑的、任何环境变量的规范。依据特权策略，对声明的特权，包括那些来自第 16.2 节的角色程序、第 16.3 节的委托程序以及任何相关的环境变量（例如，以天计的时间或当前的账目平衡），进行比较，以确定它们对使用范畴而言是否足够。如果 **acceptablePrivilegePolicies** 扩展出现，那么若特权验证者进行比较所依据的特权策略为本扩展中所含的策略之一，则特权声明只能成功。

如果成功完成比较，那么任何相关的用户通知都将提供给特权验证者。

16.2 角色处理程序

如果声明的证书是一个角色指派证书，那么特权验证者将获得指派给该角色的特定特权。特权声明者指派的角色名称包含在证书的 **role** 属性中。如果特权声明者尚未用命名角色的特权进行配置，那么可能需要确定角色规范证书，以便将特权指派给该角色。**role** 属性中和 **roleSpecCertIdentifier** 扩展中的信息可用于确定该证书。

指派给角色的特权隐含地指派给特权声明者，并因此包括在所声明的特权中，依据第 16.1 节基本程序中的特权策略对其进行比较，以确定所声明的特权对使用范畴而言是否足够。

16.3 委托处理程序

如果所声明的特权通过一个中间 AA 来委派给特权声明者，那么特权声明者将通过确保以下内容，来确保通路是一条有效的委托通路：

— 授权在委托通路中发放证书的每个 AA 这么做；

— 对施加于其上的通路和名称约束而言，委托通路中的每个证书都是有效的；

— 利用公开密钥证书（依据任何所施加的策略约束，证书都是有效的）对委托通路中的每个实体进行鉴权；

— 任何 AA 委托特权都不大于该 AA 所持有的特权。

在开始委托通路验证之前，特权验证者将获得以下内容。任何这些内容都可以由特权声明者来提供，或由特权验证者获自某个其他渠道，例如号码簿。可以在一个结构化的文档中或通过某种其他方式将服务的各属性提供给特权验证者。

— 在公开确认密钥中建立信任，用于验证信任 SOA 的签名。可以通过带外方式或通过 CA 向 SOA 发放的一个公开密钥证书（特权验证者已在其中建立信任）来建立这种信任。这样一个证书将包含 **sOAIdentifier** 扩展。

— 特权声明者的特权，在其属性证书中或其公开密钥证书的对象号码簿属性扩展中进行编码。

— 从特权声明者到信任 SOA 的证书委托通路。

— 对所声明特权的控制规则；这可以从负责所议属性的 SOA 发放的属性描述符获得，或者可以通过带外方式获得。

— 特权策略；这可以从号码簿获得，或者通过某种带外方式获得。

— 环境变量，包括如当前日期/时间、当前的账目平衡等。

实施方案功能上将等同于源自该程序的外部行为；不过，用于从给定输入获得正确输出的特殊实施方案其所用算法不是标准化的。

在由间接发放者（DS）发放属性证书的情况下，信赖方应对委托链进行彻底验证，如下所述：

i） 开始于终端实体 AC，RP 抽取发放者名称和 issuedOnBehalfOf 名称。

ii） RP 检索发放者的 AC，并验证发放者是否为 SOA 的一个间接发放者（即看它是否拥有 indirectIssuer 扩展）。

iii） RP 检索 issuedOnBehalfOf AA 的 AC，并验证 AA 是否拥有一个发放给终端实体的特权属性的超级。

RP 利用 AA 的 AC 递归步骤 ii），并因此对链进行提升，直至 SOA 所发放之 AA 的 AC。

16.3.1 验证控制规则的完整性

控制规则与委托的特权相关。获得控制规则的语法和方法不是标准化的。不过，可以对检索到的控制规则的完整性进行验证。负责所委托属性的 SOA 发放的属性描述符证书可以包含一个有关控制规则的散列。特权验证者可以对检索到的控制规则拷贝复制散列函数，并对两个散列进行比较。如果它们相同，那么特权验证者拥有准确的控制规则。

16.3.2 建立有效的委托通路

特权验证者将寻找委托通路，并获得通路中每个实体的证书。委托通路从直接特权声明者延伸至 SOA。委托通路中的每个中间证书将包含 **basicAttConstraints** 扩展，其机构部件设为 **TRUE**。每个证书的发放者将等同于证书的持有者/对象，在委托通路中二者邻近。**authorityAttributeIdentifier** 扩展用于确定委托通路中邻近实体的适当证书。从每个实体到直接特权声明者（包括）通路中的证书数量减去实体 **basicAttConstraints** 扩展中 **pathLenConstraint** 值的值不得大于 2。这是因为 **pathLenConstraint** 限制了两个端点之间的中间证书数量（即包含约束的证书以及终端实体证书），因此最大长度为该约束的值加上作为端点的证书。

如果 **delegatedNameConstraints** 扩展出现在委托通路的任何证书中，那么将以第 10 节中认证通路处理程序中处理 **nameConstraints** 扩展的相同方式，对约束进行处理。

如果 **acceptableCertPolicies** 扩展出现在委托通路的任何证书中，那么特权验证者将确保利用包含至少一个可接受策略的公开密钥证书，来对委托通路中的每个后续实体进行鉴权。

16.3.3 验证特权委托

任何委托者都不能委托比其所拥有之特权大的特权。属性描述符属性中的控制规则提供了当给定值“小于”所委托属性另一个值时的规则。

对委托通路中的每个证书，包括直接特权声明者的证书，特权验证者将确保委托者有权委托其所拥有的特权，并确保所委托的特权不大于所拥有的特权。

对这些证书中的每个证书，特权验证者都将依据有关特权的控制规则，对委托的特权和委托者拥有的特权进行比较。委托者所拥有的特权获自委托通路中的邻近证书，如第 16.2 节所述。基于第 16.3.1 节中所述的控制规则对两个特权进行比较。

16.3.4 通过/未通过决定

假设建立了一条有效的委托通路，那么直接特权声明者的特权作为以下比较的输入提供，即依据第 16.1 节中所述的特权策略进行比较，以确定直接特权声明者对使用范畴而言是否拥有足够的特权。

17 PMI号码簿方案

本节定义了用于描述号码簿中 PMI 信息的号码簿方案元素。它包括相关对象类别、属性和属性值匹配规则的规范。

17.1 PMI号码簿对象类别

本子节定义了用于描述号码簿中 PMI 对象的对象类别定义。

17.1.1 PMI用户对象类别

在定义对象条目过程中使用 PMI 用户对象类别，它可以是属性证书的持有者。

```
pmiUser OBJECT-CLASS ::= {
-- 一个 PMI 用户（即一个“持有者”）
  SUBCLASS OF     {top}
  KIND            auxiliary
  MAY CONTAIN     {attributeCertificateAttribute}
  ID              id-oc-pmiUser }
```

17.1.2 PMI AA对象类别

在定义对象条目过程中使用 PMI AA 对象类别，它作为属性机构。

```
pmiAA OBJECT-CLASS ::= {
-- 一个 PMI AA
  SUBCLASS OF     {top}
  KIND            auxiliary
  MAY CONTAIN     {aACertificate |
                   attributeCertificateRevocationList |
                   attributeAuthorityRevocationList}
  ID              id-oc-pmiAA }
```

17.1.3 PMI SOA对象类别

在定义对象条目过程中使用 PMI SOA 对象类别，它作为机构源。注意：如果通过发放一个包含 **sOAIdentifier** 扩展的公开密钥证书，授权对象作为一个 SOA，那么代表对象的号码簿条目也将包含 **pkiCA** 对象类别。

```
pmiSOA OBJECT-CLASS ::= {  -- 一个 PMI 机构源
  SUBCLASS OF     {top}
  KIND            auxiliary
  MAY CONTAIN     {attributeCertificateRevocationList |
                   attributeAuthorityRevocationList |
                   attributeDescriptorCertificate}
  ID              id-oc-pmiSOA }
```

17.1.4 属性证书CRL分发点对象类别

在定义对象条目过程中使用属性证书 CRL 分发点对象类别，它包含属性证书与/或属性机构撤消清单片段。本辅助类别旨在当实例化条目时实现与 **crlDistributionPoint** 结构对象类别的结合。由于在这种情况下 **certificateRevocationList** 和 **authorityRevocationList** 属性是可选的，因此有可能创建例如只包含一个属性机构撤消清单的条目，或者创建包含多类型撤消清单的条目，这取决于不同要求。

```
attCertCRLDistributionPt     OBJECT-CLASS ::= {
  SUBCLASS OF     {top}
  KIND            auxiliary
  MAY CONTAIN     { attributeCertificateRevocationList |
                    attributeAuthorityRevocationList }
  ID              id-oc-attCertCRLDistributionPts }
```

17.1.5 PMI委托通路

在定义对象条目过程中使用 PMI 委托通路对象类别，它可以包含委托通路。通常它将与结构化的对象类别 **pmiAA** 的条目结合使用。

```
pmiDelegationPath       OBJECT-CLASS      ::= {
    SUBCLASS OF         {top}
    KIND                auxiliary
    MAY CONTAIN         { delegationPath }
    ID                  id-oc-pmiDelegationPath }
```

17.1.6 特权策略对象类别

在定义对象条目过程中使用特权策略对象类别，它包含特权策略信息。

```
privilegePolicy       OBJECT-CLASS ::= {
    SUBCLASS OF       {top}
    KIND              auxiliary
    MAY CONTAIN       {privPolicy }
    ID                id-oc-privilegePolicy }
```

17.1.7 受保护的特权策略对象类别

在定义对象条目过程中使用受保护的特权策略对象类别，它包含属性证书中受保护的特权策略。

```
protectedPrivilegePolicy          OBJECT-CLASS      ::= {
    SUBCLASS OF       {top}
    KIND              auxiliary
```

```
    MAY CONTAIN         {protPrivPolicy }
    ID                  id-oc-protectedPrivilegePolicy }
```

17.2 PMI号码簿属性

本子节定义用于在号码簿条目中保存 PMI 数据的号码簿属性。

17.2.1 属性证书属性

以下属性包含发放给某个特定持有者的各属性证书，并保存在该持有者的号码簿条目中。

```
attributeCertificateAttribute ATTRIBUTE  ::= {
    WITH SYNTAX                 AttributeCertificate
    EQUALITY MATCHING RULE      attributeCertificateExactMatch
    ID                          id-at-attributeCertificate }
```

17.2.2 AA证书属性

以下属性包含发放给某个 AA 的各属性证书，并保存在持有者 AA 的号码簿条目中。

```
aACertificate       ATTRIBUTE   ::={
    WITH SYNTAX                 AttributeCertificate
    EQUALITY MATCHING RULE      attributeCertificateExactMatch
    ID                          id-at-aACertificate }
```

17.2.3 属性描述符证书属性

以下属性包含由某个 SOA 发放的各属性证书，它们包含 **attributeDescriptor** 扩展。这些属性证书包含有效的语法和特权属性的控制规则说明，并保存在发放 SOA 的号码簿条目中。

```
attributeDescriptorCertificate      ATTRIBUTE  ::= {
     WITH SYNTAX                        AttributeCertificate
     EQUALITY MATCHING RULE             attributeCertificateExactMatch
     ID                                 id-at-attributeDescriptorCertificate }
```

17.2.4 属性证书撤消清单属性

以下属性包含一个有关撤消属性证书的清单。这些清单可以保存在发放机构的号码簿条目中，或者保存在其他的号码簿条目中（例如，一个分发点）。

```
attributeCertificateRevocationList      ATTRIBUTE ::= {
    WITH SYNTAX                         CertificateList
    EQUALITY MATCHING RULE              certificateListExactMatch
    ID                                  id-at-attributeCertificateRevocationList}
```

17.2.5 AA证书撤消清单属性

以下属性包含一个有关发放给 AA 的撤消属性证书的清单。这些清单可以保存在发放机构的号码簿条目中，或者保存在其他的号码簿条目中（例如，一个分发点）。

```
attributeAuthorityRevocationList        ATTRIBUTE ::= {
    WITH SYNTAX                         CertificateList
    EQUALITY MATCHING RULE              certificateListExactMatch
    ID                                  id-at-attributeAuthorityRevocationList }
```

17.2.6 委托通路属性

委托通路属性包含各委托通路，每个委托通路由一系列属性证书组成。

```
delegationPath      ATTRIBUTE       ::= {
    WITH SYNTAX             AttCertPath
    ID                      id-at-delegationPath }

    AttCertPath     ::=   SEQUENCE OF AttributeCertificate
```

本属性可以保存在 AA 号码簿条目中，并将包含一些从该 AA 到其他 AA 的委托通路。如果使用，本属性将使对受委托属性证书的检索更加高效，这些证书形成常用的委托通路。这样，对将要用的本属性就没有任何特殊的要求，并且保存在属性中的值集将不可能代表有关任何特定 AA 的完整的委托通路集。

17.2.7 特权策略属性

特权策略属性包含有关特权策略的信息。

```
privPolicy ATTRIBUTE      ::= {
    WITH SYNTAX           PolicySyntax
    ID                    id-at-privPolicy }
```

policyIdentifier 部件包括为特殊特权策略注册的对象标识符。

如果 **content** 出现，那么将包括完整的特权策略内容。

如果 **pointer** 出现，那么 **name** 部件将参考一个或多个位置，在这些位置可以找到特权策略的拷贝。如果 **hash** 部件出现，那么它将包含一个有关特权策略内容的散列，应在所参考的位置上能找到它。可用该散列对所参考文档的完整性进行检查。

17.2.8 受保护的特权策略属性

受保护的特权策略属性包含在属性证书中受保护的特权策略。

```
protPrivPolicy      ATTRIBUTE      ::= {
WITH SYNTAX                        AttributeCertificate
EQUALITY MATCHING RULE             attributeCertificateExactMatch
ID                                 id-at-protPrivPolicy }
```

注意：不像典型的属性证书，**protPrivPolicy** 属性内的那些证书包含的是特权策略而不是特权。这些属性证书的发放者和持有者部件确定相同的实体。包含在 **protPrivPolicy** 属性内的属性证书所包括的属性为 **privPolicy** 属性或 **xmlPrivPolicy** 属性。

17.2.9 XML保护的特权策略属性

XML 保护的特权策略属性包含 XML 编码的特权策略信息。

```
xmlPrivPolicy      ATTRIBUTE      ::= {
WITH SYNTAX        UTF8String -- 包含 XML 编码的特权策略信息
ID                 id-at-xMLPprotPrivPolicy }
```

17.3 PMI普通号码簿匹配规则

本子节定义 PMI 号码簿属性的匹配规则。

17.3.1 属性证书准确匹配

属性证书准确匹配规则对出现的值与类型 **AttributeCertificate** 的属性值进行比较，看它们是否相同。

```
attributeCertificateExactMatch MATCHING-RULE  ::= {
    SYNTAX          AttributeCertificateExactAssertion
    ID              id-mr-attributeCertificateExactMatch }

AttributeCertificateExactAssertion    ::=       SEQUENCE {
    serialNumber            CertificateSerialNumber,
    issuer                  AttCertIssuer }
```

如果属性值中的各部件匹配于出现值中的那些部件，那么本匹配规则返回 TRUE。

17.3.2 属性证书匹配

属性证书匹配规则对出现的值与类型 **AttributeCertificate** 的属性值进行比较。本匹配规则允许进行比 **certificateExactMatch** 更复杂的匹配。

```
attributeCertificateMatch  MATCHING-RULE  ::=  {
    SYNTAX          AttributeCertificateAssertion
    ID              id-mr-attributeCertificateMatch  }

AttributeCertificateAssertion  ::=  SEQUENCE  {
    holder          [0]    CHOICE {
                           baseCertificateID    [0]  IssuerSerial,
                              holdertName               [1]  GeneralNames} OPTIONAL,
    issuer          [1]    GeneralNames OPTIONAL,
    attCertValidity [2]    GeneralizedTime OPTIONAL,
    attType         [3]    SET OF AttributeType OPTIONAL}
```

-- 至少应出现序列的一个部件。

如果出现在出现值中的所有部件都匹配于属性值的各对应部件，那么本匹配规则返回 TRUE，如下所示：

— 如果它等于保存属性值的 **IssuerSerial** 部件，那么 **baseCertificateID** 匹配；

— 如果保存属性值包含以下名称扩展，即其名称类型与出现值中所述的名称类型相同，那么 **holderName** 匹配；

— 如果保存属性值包含以下名称部件，即其名称类型与出现值中所述的名称类型相同，那么 **issuer** 匹配；

— 如果它落在保存属性值规定的有效周期内，那么 **attCertValidity** 匹配；以及

— 对出现值中的每个 **attType**，有一个该类型的属性，出现在保存值的 **attributes** 部件中。

17.3.3 持有者发放者匹配

属性证书持有者发放者匹配规则对出现值持有者与/或发放者的出现值与类型 **AttributeCertificate** 的属性值进行比较，看它们是否相同。

```
holderIssuerMatch MATCHING-RULE ::= {
    SYNTAX          HolderIssuerAssertion
    ID              id-mr-holderIssuerMatch }

HolderIssuerAssertion    ::=    SEQUENCE {
    holder          [0]          Holder          OPTIONAL,
    issuer          [1]          AttCertIssuer   OPTIONAL }
```

如果出现在出现值中的所有部件都匹配于属性值的各对应部件，那么本匹配规则返回 TRUE。

17.3.4 委托通路匹配

delegationPathMatch 匹配规则对出现的值与类型 **delegationPath** 的属性值进行比较，看它们是否相同。特权验证者可以使用本匹配规则来选择一条开始于其 SOA 发放的证书并结束于发放给 AA 的证书的通路，AA 发放正在接受验证的终端实体持有者证书。

```
delegationPathMatch MATCHING-RULE ::= {
    SYNTAX          DelMatchSyntax
    ID              id-mr-delegationPathMatch }

DelMatchSyntax    ::=  SEQUENCE {
    firstIssuer     AttCertIssuer,
    lastHolder      Holder }
```

如果 **firstIssuer** 部件中出现的值匹配于保存值中 **SEQUENCE** 中第一个证书的发放者字段的对应元素，以及 **lastHolder** 部件中出现的值匹配于保存值中 **SEQUENCE** 中最后一个证书的持有者字段的对应元素，那么本匹配规则返回 TRUE。如果二者都匹配失败，那么本匹配规则返回 FALSE。

第 4 部分 — 号码簿使用公开密钥与属性证书框架

号码簿使用公开密钥证书框架作为诸多安全服务的基础，包括强鉴权和号码簿操作的保护，以及已保存数据的保护。号码簿使用属性证书框架作为基于规则的访问控制方案的基础。此处定义了公开密钥证书框架元素和属性证书框架元素与各种号码簿安全服务之间的关系。通过完整的号码簿规范集对号码簿提供的特定安全服务做了全面规定。

18 号码簿鉴权

号码簿为用户和其他 DSA 提供用户（通过 DUA 访问号码簿）鉴权和号码簿系统（DSA）鉴权。依据环境，可以使用简单鉴权或强鉴权。号码簿中用于简单鉴权和强鉴权的程序在下面各子节中进行描述。

18.1 简单的鉴权程序

简单鉴权旨在基于用户的不同名称、双方商定的（可选）口令、单个域内双方对该口令使用和处理方式的理解提供本地授权。简单鉴权的使用主要面向本地使用，即一个 DUA 和一个 DSA 之间或一个 DSA 和一个 DSA 之间的对等实体鉴权。简单鉴权可以通过以下几种方式实现：

a） 将用户不同的名称和（可选的）明文形式的口令（未做保护的）传送给接收者，用于评估；

b） 传送用户不同的名称、口令、随机数与/或时戳，所有这些都利用单路函数进行了保护；

c） 传送 b）中所述的保护信息以及随机数与/或时戳，所有这些都利用单路函数进行了保护。

注 1 — 不要求所用的单路函数是不同的。

注 2 — 保护口令程序的信令可以是一个文档扩展问题。

当口令不受保护时，为防止未经授权访问而提供的安全等级最低。不应将之看作是安全服务的基础。对用户不同的名称和口令进行保护提供了更高等级的安全。典型地，用于保护机制的算法为非加密、单路函数，它们非常容易实现。

实现简单鉴权的一般程序如图 5 所示。

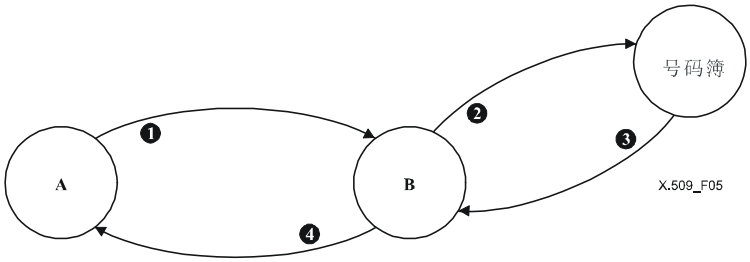


图 5－无保护的简单鉴权程序

涉及以下步骤：

1） 发起用户 A 将其不同的名称和口令发送给接收用户 B；

2） B 将 A 假设的不同名称和口令发送给号码簿，依据有关 A 的号码簿条目中所持的 **UserPassword** 属性，对口令进行检查（使用号码簿的比较操作）；

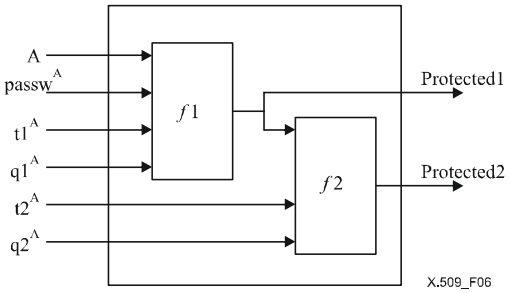
3） 号码簿向 B 确认（否认）证书是有效的；

4） 可以将鉴权成功（或失败）的信息传送给 A。

最基本的简单鉴权形式只涉及步骤 1），在 B 已经检查了不同的名称和口令后，可以包括步骤 4）。

18.1.1 产生受保护的鉴别信息

图 6 描述了两种方法，利用它们可以产生受保护的鉴别信息。*f*1 和 *f*2 为单路函数（相同的或不同的），时戳和随机数是可选的，并依据双边协议。



A	用户的不同名称
t^A	时戳
$Passw^A$	A的口令
q^A	随机数，可选地可以包括一个计数器

图 6－受保护的简单鉴权

18.1.2 受保护的简单鉴权程序

图 7 描述了受保护的简单鉴权程序。

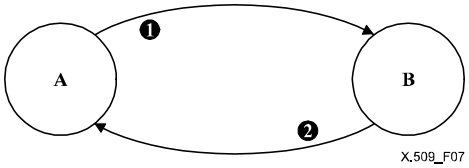


图 7－受保护的简单鉴权程序

涉及以下步骤（最初只使用 *f*1）：

1） 发起用户 A 将其受保护的鉴别信息（鉴权者 1）发送给用户 B。通过应用图 6 的单路函数（*f*1）来实现保护功能，其中的时戳与/或随机数（当使用的时候）用于最大可能地减少重播和隐藏口令。

关于 A 的口令的保护如下式所示：

$$Protected1 = f1\ (t1^A, q1^A, A, passw^A)$$

传送给 B 的信息如下式所示：

$$Authenticator1 = t1^A, q1^A, A, Protected1$$

2） 通过产生一个本地的、受保护的 A 口令拷贝（以 Protected1 形式），B 对 A 提供的受保护鉴别信息进行验证（利用 A 提供的不同名称和可选时戳与/或随机数，以及 A 口令的本地拷贝）。B 对假设的鉴别信息（Protected1）和本地产生的值进行比较，看它们是否相同。

3） B 向 A 确认（否认）通过对受保护鉴别信息的验证；

可以对程序进行修改，以便利用 *f*1 和 *f*2 提供更大的保护。主要的区别如下所示：

1） A 将其额外保护的鉴别信息（Authenticator2）发送给 B。通过应用一个更进一步的单路函数 *f*2 来实现额外的保护，如图 6 所示。更进一步的保护如下式所示：

$$Protected2 = f2\ (t2^A, q2^A, Protected1)$$

传送给 B 的信息如下式所示：

$$Authenticator2 = t1^A, t2^A, q1^A, q2^A, A, Protected2$$

为了进行比较，B 产生一个 A 额外保护口令的本地值，并将其与 Protected2 的值进行比较，看它们是否相同。

2） B 向 A 确认（否认）通过对受保护鉴别信息的验证。

注 — 在这些节中定义的各程序是依据 A 和 B 规定的。当应用于号码簿时（在 ITU-T X.511 建议书 | ISO/IEC 9594-3 和 ITU-T X.518 建议书 | ISO/IEC 9594-4 中规定），A 可以是一个绑定于一个 DSA，B 的 DUA；可选地，A 也可以是一个绑定于另一个 DSA，B 的 DSA。

18.1.3 用户口令属性类型

用户口令属性类型包含一个对象的口令。用户口令的属性值是一个由该对象规定的字符串。

```
userPassword    ATTRIBUTE   ::=   {
    WITH SYNTAX                   OCTET STRING (SIZE (0..ub-user-password))
    EQUALITY MATCHING RULE        octetStringMatch
    ID                            id-at-userPassword }
```

18.2 强鉴权

本子节中所述的程序用于一个 DUA 与一个 DSA 之间的鉴权，以及 DSA 对之间的鉴权。程序使用本规范中定义的公开密钥框架。另外，程序还利用号码簿本身作为实施鉴权所需公开密钥信息的库。在协议规范本身中定义需要在号码簿协议中包括的相关参数。除了号码簿，使用这样一个库的应用也可以使用在此定义的程序。为了号码簿使用这些程序，在这些程序中的术语“用户”可以指一个 DUA 或一个 DSA。

本号码簿规范中所用的强鉴权方法使用了一个加密系统族的特性，即公开密钥加密系统（PKCS）。这些加密系统，也描述为非对称的加密系统，涉及一对密钥，一个密钥是专用的，一个密钥是公开的，而不像在传统的加密系统中，只有一个单个密钥。附件 E 对这些加密系统以及使之在鉴权中发挥作用的各特性做了简介。对当前在本鉴权框架中有用的 PKCS，它应拥有以下特性，即密钥对中的两个密钥都能用于加密，如果使用公开密钥，那么使用专用密钥来解密，如果使用专用密钥，那么使用公开密钥来解密。换言之，$X_p \cdot X_s = X_s \cdot X_p$，其中 X_p/X_s 为使用用户 X 公开/专用密钥的加密/解密函数。

注 — 可选的 PKCS 类型可能是未来的一种扩展，即不要求具有交换性特性的类型，并且无需对本号码簿规范做大的改动即可支持。

本鉴权框架不要求使用某个特殊的加密系统。意指框架适用于任何合适的公开密钥加密系统，并因此支持对方法的修改，作为未来密码系统、数学技术或计算能力发展的结果。不过，为了正确实施鉴权，希望进行鉴权的两个用户需支持相同的加密算法。因此，在一系列相关应用的范畴内，单个算法的选择应尽可能使更多的用户团体能够实现可靠的鉴权和通信。

鉴权依赖于每个用户拥有一个惟一的不同名称。由命名机构负责分配不同的名称。每个用户因此相信命名机构不会发放重复的不同名称。

通过看它是否拥有其专用密钥来确定每个用户。第二个用户能够确定通信伙伴是否拥有专用密钥，并能利用它来验证通信伙伴实际上就是用户。验证的有效性依赖于专用密钥对用户是保密的。

为了一个用户确定某个通信伙伴是否拥有另一个用户的专用密钥，它自身应拥有该用户的公开密钥。当直接从号码簿中用户条目获得该公开密钥的值时，验证其正确与否更成问题。对此，有许多可能的方法：子节 18.2.1 描述了一个过程，利用它，参考号码簿，可以对用户的公开密钥进行检查。只有当需要鉴权的用户之间存在一条未断的号码簿信任点链时，本过程才能工作。可以通过确定一个公共的信任点来构造这样一条链。通过一条未断的信任点链将该信任公共点连至各用户。

18.2.1 从号码簿获取公开密钥证书

证书保存在号码簿条目中，作为类型 **UserCertificate**、**CACertificate** 和 **CrossCertificatePair** 的属性。号码簿知道这些属性类型。可以使用与其他属性相同的协议操作来操作这些属性。可以在第 3.3 中找到这些类型的定义；在第 11.2 节中对这些属性类型的规范进行定义。

通常，在用户可以相互鉴权之前，号码簿将提供完整的认证，并返回认证通路。不过，实际上，对某个特殊的鉴权实例，可以通过以下方式，减少需要从号码簿中获得的信息量：

a） 如果由相同的认证机构来为想鉴权的两个用户提供服务，那么认证通路将变得不那么重要，各用户直接展开彼此的证书；

b） 如果在层次结构中安排用户的 CA，那么用户可以保存公开密钥，以及用户与 DIT 根之间所有认证机构的证书和逆证书。典型地，这将涉及知道公开密钥以及仅三个或四个认证机构的用户。而后用户只需从公共信任点获得认证通路；

c） 如果一个用户常与由某个特殊的其他 CA 认证的用户进行通信，那么该用户可以知道至该 CA 的认证通路以及自该 CA 的返回认证通路，使之只需从号码簿获得该其他用户自身的证书；

d） 认证机构可以通过双边协议相互间进行交叉认证。结果是缩短了认证通路；

e） 如果两个用户之前已经进行了通信，并知道了彼此的证书，那么无需借助号码簿，它们就能够进行鉴权。

任何情况下，当从认证通路相互间知道证书后，各用户将对接收到的证书的有效性进行检查。

18.2.1.1 举例

图 8 描述了一个有关 DIT 分段的假设例子，图中的 CA 形成一个层次结构。除了 CA 上所示的信息，我们还假定每个用户都知道其认证机构的公开密钥以及其自身的公开密钥和专用密钥。

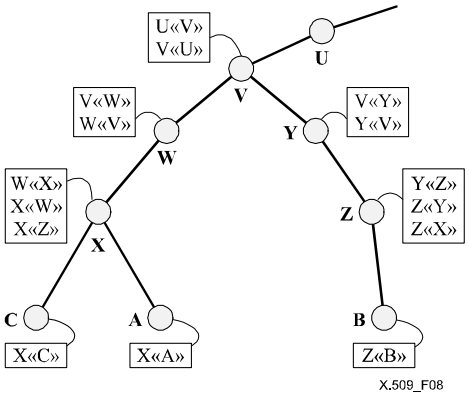


图 8－CA层次结构 — 一个假设的例子

如果用户的 CA 安排在层次结构中，那么 A 可以从号码簿中获得以下证书，以建立一条到 B 的认证通路：

$$X\langle\langle W\rangle\rangle, W\langle\langle V\rangle\rangle, V\langle\langle Y\rangle\rangle, Y\langle\langle Z\rangle\rangle, Z\langle\langle B\rangle\rangle$$

当 A 已获得这些证书时，它可依次展开认证通路，以获得 B 证书的内容，包括 Bp：

$$Bp = Xp \bullet X\langle\langle W\rangle\rangle\ W\langle\langle V\rangle\rangle\ V\langle\langle Y\rangle\rangle\ Y\langle\langle Z\rangle\rangle\ Z\langle\langle B\rangle\rangle$$

通常，A 还需要从号码簿获得以下证书，以便建立从 B 到 A 的返回认证通路：

Z<<Y>>, Y<<V>>, V<<W>>, W<<X>>, X<<A>>

当 B 从 A 收到这些证书时，它可依次展开返回认证通路，以获得 A 证书的内容，包括 Ap：

Ap = Zp • Z<<Y>> Y<<V>> V<<W>> W<<X>> X<<A>>

对第 18.2.1 节进行最优化：

a） 取 A 和 C，例如：二者都知道 Xp，因此 A 只需简单地直接获得 C 的证书。展开证书通路简化为：

Cp = Xp • X<<C>>

以及展开返回认证通路简化为：

Ap = Xp • X<<A>>

b） 假设 A 因此将知道 W<<X>>、Wp、V<<W>>、Vp、U<<V>>、Up 等，简化 A 需要从号码簿中获得的信息，以形成认证通路：

V<<Y>>, Y<<Z>>, Z<<B>>

以及 A 需要从号码簿中获得的信息，以形成返回认证通路：

Z<<Y>>, Y<<V>>

c） 假设 A 常与通过 Z 进行认证的用户进行通信，那么它可以知道（除了在上面 b）中知道的公开密钥之外）V<<Y>>、Y<<V>>、Y<<Z>>，以及 Z<<Y>>。为了实现与 B 的通信，它只需要从号码簿获得 Z<<B>>。

d） 假设通过 X 进行认证的用户与通过 Z 进行认证的用户经常进行通信，那么将在有关 X 的号码簿条目中持有 X<<Z>>，反之亦然（如图 8 所示）。如果 A 想对 B 进行鉴权，那么 A 只需获得：

X<<Z>>, Z<<B>>

以形成认证通路，以及：

Z<<X>>

以形成返回认证通路。

e） 假设用户 A 和用户 C 之前已进行通信，并已知道彼此间的证书，那么它们可以直接使用彼此间的公开密钥，即：

Cp = Xp • X<<C>>

以及

Ap = Xp • X<<A>>

在更一般情况下，认证机构与层次结构方式无关。指的是图 9 中的层次结构例子，假设一个通过 U 进行认证的用户 D，希望对通过 W 进行认证的用户 E 进行鉴权。用户 D 的号码簿条目将持有证书 U<<D>>，用户 E 的条目将持有证书 W<<E>>。

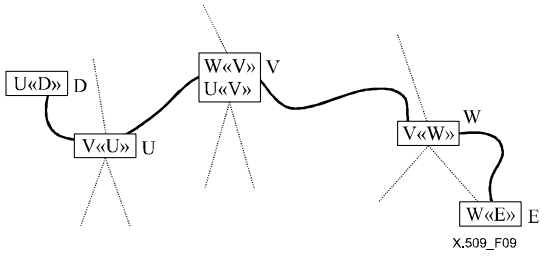


图 9－非层次型认证通路 — 一个例子

设 V 为一个 CA，利用它，CA U 和 W 在之前的某个时候以一种信任的方式对公开密钥进行了交换。结果是，产生了证书 U<<V>>、V<<U>>、W<<V>>和 V<<W>>，并保存在号码簿中。假设 U<<V>>和 W<<V>>保存在 V 的条目中，V<<U>>保存在 U 的条目中，V<<W>>保存在 W 的条目中。

用户 D 需要找到一条到 E 的认证通路。可以使用各种不同的策略。一种策略是将用户和 CA 看作是节点，证书是有向图中的弧。在这些项中，D 需要在图中执行搜索，以便找到一条从 U 到 E 的通路，如 U<<V>>、V<<W>>、W<<E>>。当找到该通路后，还可以构造逆通路 W<<V>>、V<<U>>、U<<D>>。

18.2.2 强鉴权程序

上面对基本的鉴权方法进行了描述，即通过验证拥有专用密钥来对身份进行验证。不过，可能有许多使用本方法的鉴权程序。通常由一个特定的应用来确定适当的程序，以便满足应用的安全策略要求。本节描述三个特殊的鉴权程序，可以发现，它们对一系列应用都是有用的。

注 — 本号码簿规范对各程序的规定不详细到实施所需的详细程度。不过，可以设想额外的标准，它们可以详细到这样的程度，或者以某个应用特定的方式，或者以一般性目的的方式。

三个程序涉及不同数量的鉴权信息交换，并因此为其参与者提供了不同类型的保证。尤其是：

a） 单路鉴权，在第 18.2.2.1 节中描述，它涉及一个从一个用户（A）到另一个用户（B）的单个信息传送，并建立以下内容：

— A 的身份，以及 A 实际产生的鉴权令牌；

— B 的身份，以及实际计划发送给 B 的鉴权令牌；

— 所传送鉴权令牌的完整性和“独创性”（未被传送两次或更多次的特性）。

可以为伴随传送的任意附加数据建立后面的特性；

b） 双路鉴权，在第 18.2.2.2 节中描述，它另外涉及一个从 B 到 A 的答复。它另外建立以下内容：

— 在答复中产生的鉴权令牌实际上是由 B 产生的，并计划发送给 A；

— 在答复中发送的、有关鉴权令牌的完整性和独创性；

— （可选地）相互保密的令牌部分；

c） 三路鉴权，在第 18.2.2.3 节中描述，它另外涉及一个从 A 到 B 的传送。它建立与双路鉴权相同的特性，但这么做时无需与时戳检查相关联。

对进行强鉴权的每种情况，A 将获得 B 的公开密钥，并在进行任何信息交换之前，从 B 返回认证通路给 A。这可能涉及对号码簿的访问，如第 18.2 节所述。在下面的程序描述中，将不再提及任何此类访问。

在下面节中所述的时戳检查仅应用于以下场合，即当在本地环境中使用同步时钟时，或者如果通过双边协议来实现各时钟的逻辑同步。在任何情况下，都建议使用协同的世界时间。

对下面所述的三个鉴权程序中的每一个，均假设 A 方已对认证通路中所有证书的有效性进行了检查。

18.2.2.1 单路鉴权

涉及以下步骤，如图 10 所示：

1） A 产生 r^A，一个非重复的数字，用于检测重播攻击并防止伪造；

2） A 向 B 发送以下消息：

$$B \quad A,\ A\{t^A,\ r^A,\ B\}$$

其中 t^A 是一个时戳。t^A 由一个或两个日期组成：令牌的产生时间（它是可选的）和终止时间。可选地，如果“sgnData”的数据源鉴权由数字签名提供：

$$BA, A\{t^A, r^A, B, sgnData\}$$

在对信息进行传送的情况下，传送的信息之后将用作为一个专用密钥（该信息指的是“encData”）：

$$BA, A\{t^A, r^A, B, sgnData, Bp[encData]\}$$

那么将“encData”用做一个专用密钥意味着应对它进行仔细选择，例如，对任何所用的加密系统而言都是一个强密钥，这在令牌的“sgnData”字段中指出。

3） B 完成以下操作：

a） 从 BA 获得 Ap，检查 A 的证书是否已过期；

b） 检查签名，而后检查经签署信息的完整性；

c） 检查 B 本身是否为预期的接收者；

d） 检查时戳是否为“当前的”；

e） 可选地，检查 r^A 未被重播。例如，这可以通过使 r^A 包括一个连续的部分来实现，通过一个本地实施方案来对连续部分值的惟一性进行检查。

在 t^A 所指的过期日期之前，r^A 都是有效的。r^A 总是伴随一个连续的部分，它指明在时间范围 t^A 期间 A 不得重复令牌，因此不需要对 r^A 本身的值进行检查。

任何情况下，在时间范围 t^A 期间，B 方保存连续部分以及明确的时戳 t^A 和令牌的散列部分都是合理的。

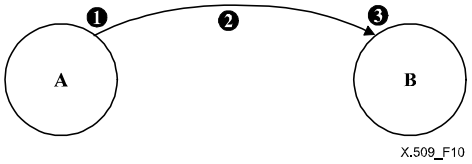


图 10－单路鉴权

18.2.2.2 双路鉴权

涉及以下步骤，如图 11 所示：

1） 对第 18.2.2.1 节；

2） 对第 18.2.2.1 节；

3） 对第 18.2.2.1 节；

4） B 产生 r^B，一个非重复的数字，用于类似于 r^A 的目的；

5） B 向 A 发送以下鉴权令牌：

$$B\{t^B, r^B, A, r^A\}$$

其中 t^B 是以与 t^A 相同的方式定义的一个时戳；

可选地，如果“sgnData”的数据起源鉴权由数字签名提供：

$$B\{t^B, r^B, A, r^A, sgnData\}$$

在对信息进行传送的情况下，传送的信息之后将用作为一个专用密钥（该信息指的是“encData”）：

$$B\{t^B, r^B, A, r^A, sgnData, Ap[encData]\}$$

那么将“encData”用做一个专用密钥意味着应对它进行仔细选择，例如，对任何所用的加密系统而言都是一个强密钥，这在令牌的“sgnData”字段中指出。

6） A 完成以下操作：

a） 验证签名，而后验证经签署信息的完整性；

b） 检查 A 是否为预期的接收者；

c） 检查时戳 t^B 是否为“当前的”；

d） 可选地，检查 r^B 未被重播（参见第 18.2.2.1 节，步骤 3），d））。

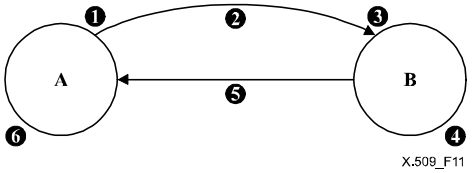


图 11－ 双路鉴权

18.2.2.3 三路鉴权

涉及以下步骤，如图 12 所示：

1） 对第 18.2.2.2 节；

2） 对第 18.2.2.2 节；时戳 t^A 可以为 0；

3） 对第 18.2.2.2 节，时戳无需检查除外；

4） 对第 18.2.2.2 节；

5） 对第 18.2.2.2 节；时戳 t^B 可以为 0；

6） 对第 18.2.2.2 节，时戳无需检查除外；

7） A 检查收到的 r^A 是否等同于发送的 r^A；

8） A 向 B 发送以下鉴权令牌：

$$A\{r^B,B\}$$

9） B 完成以下操作：

a） 检查签名，而后检查经签署信息的完整性；

b） 检查收到的 r^B 是否等同于 B 所发送的 r^B。

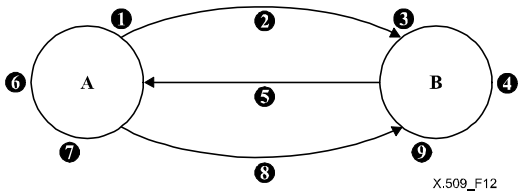


图 12－三路鉴权

19 访问控制

号码簿存在于以下环境中，即当中的不同主管部门控制着对其那部分 DIB 的访问。在号码簿范畴中，访问控制方案的定义包括用于以下操作的方法：

— 规定访问控制信息（ACI）；

— 执行该访问控制信息定义的访问权限；

— 维护访问控制信息。

执行访问权限用于控制对以下信息的访问：

— 与名称相关的号码簿信息；

— 号码簿用户信息；

— 号码簿操作信息，包括访问控制信息。

在执行其安全策略中，主管部门可以利用所有或部分任何标准化的访问控制方案，或者可以自由地自行确定其自身的方案。

在 ITU-T X.501 建议书 | ISO/IEC 9594-2 中定义的基本访问控制（BAC）方案是一个基于访问控制清单的方案，使得号码簿管理者能够将许可与所执行的鉴权级别结合起来，以便绑定于号码簿。在本规范中定义的公开密钥证书框架用于提供用于该绑定的强鉴权方案。

批注 [P144]: Page: 99
A: Basic Access Control (BAC)

在 ITU-T X.501 建议书 | ISO/IEC 9594-2 中定义的、基于规则的基本访问控制（RBAC）方案利用在本规范中定义的属性证书框架来传达在访问控制决策过程中使用的许可证属性。RBAC 还可以与 BAC 结合使用。

批注 [P145]: Page: 99
A: Rules Based Access Control (RBAC)

20 对号码簿操作的保护

在本规范中定义的公开密钥证书框架用在本建议书系列的所有号码簿协议中，可选地用于保护操作，包括请求、响应和错误。完整性保护通过发送者的数字签名以及接收者利用发送者对应的公开密钥证书对该签名进行验证来提供。私密性保护通过使用公开密钥加密来提供，它利用自预期接收者的公开密钥证书处获得的公开密钥对内容进行加密，接收者则利用其对应的专用密钥进行解密。

在协议交换中用于请求和包括保护元素的特定机制和句法，在本规范系列的各个号码簿协议中进行定义。

附 件 A

公开密钥和属性证书框架

（本附件是本建议书 | 国际标准的组成部分）

本附件包括所有以 3 种 ASN.1 模块形式包含于本号码簿规范中的 ASN.1 类型、值和信息对象类别定义：**AuthenticationFramework**、**CertificateExtensions** 和 **AttributeCertificateDefinitions**。

-- *A.1* 鉴权框架模块

```
AuthenticationFramework {joint-iso-itu-t ds(5) module(1) authenticationFramework(7) 5}
DEFINITIONS ::=
BEGIN

-- EXPORTS All --
-- 输出在该模块中规定的类型和值，用于本号码簿规范中涵盖的其他ASN.1模块，还要使用它们接入到号码簿
-- 业务的其他应用中。其他的应用可以把它们用于自己的目的，但这并不会限制为维护或改进号码簿业务所需
-- 的扩展和修改。

IMPORTS
    id-at, id-nf, id-oc, informationFramework, upperBounds, selectedAttributeTypes, basicAccessControl,
    certificateExtensions
        FROM UsefulDefinitions {joint-iso-itu-t ds(5) module(1) usefulDefinitions(0) 5}

Name, ATTRIBUTE, OBJECT-CLASS, NAME-FORM, top
    FROM InformationFramework informationFramework

ub-user-password, ub-content
    FROM UpperBounds upperBounds

UniqueIdentifier, octetStringMatch, DirectoryString{}, commonName
    FROM SelectedAttributeTypes selectedAttributeTypes

certificateExactMatch, certificatePairExactMatch, certificateListExactMatch, KeyUsage, GeneralNames,

    CertificatePoliciesSyntax, algorithmIdentifierMatch, CertPolicyId
    FROM CertificateExtensions certificateExtensions ;

-- 公开密钥证书定义 --

Certificate                     ::=     SIGNED { SEQUENCE {
    version                     [0]     Version DEFAULT v1,
    serialNumber                        CertificateSerialNumber,
    signature                           AlgorithmIdentifier,
    issuer                              Name,
    validity                            Validity,
    subject                             Name,
    subjectPublicKeyInfo                SubjectPublicKeyInfo,
    issuerUniqueIdentifier      [1]     IMPLICIT UniqueIdentifier OPTIONAL,
                                        -- 如果出现，那么版本为 v2 或 v3。
     subjectUniqueIdentifier    [2]     IMPLICIT UniqueIdentifier OPTIONAL,
                                        -- 如果出现，那么版本为 v2 或 v3。
     extensions                 [3]     Extensions OPTIONAL
                                        -- 如果出现，那么版本为 v3。 -- } }

Version                 ::=     INTEGER { v1(0), v2(1), v3(2) }

CertificateSerialNumber ::=     INTEGER
AlgorithmIdentifier             ::= SEQUENCE {
     algorithm          ALGORITHM.&id ({SupportedAlgorithms}),
     parameters         ALGORITHM.&Type ({SupportedAlgorithms}{ @algorithm}) OPTIONAL }

-- 延期定义以下信息对象集，可能是为了
-- 标准化概述或规范实施一致性声明。
-- 要求信息对象集规定一个表，以约束AlgorithmIdentifier的参数部件。

SupportedAlgorithms     ALGORITHM       ::=     { ... }
```

```
Validity                    ::=   SEQUENCE {
    notBefore       Time,
    notAfter        Time }

SubjectPublicKeyInfo        ::=   SEQUENCE {
    algorithm               AlgorithmIdentifier,
    subjectPublicKey        BIT STRING }

Time  ::=  CHOICE {
    utcTime                 UTCTime,
    generalizedTime         GeneralizedTime }

Extensions ::= SEQUENCE OF Extension
-- 对那些 SEQUENCE 中单个扩展排序显得重要的扩展而言，
-- 那些单个扩展的规范将包括有关排序重要性的规则。

Extension ::= SEQUENCE {
    extnId          EXTENSION.&id ({ExtensionSet}),
    critical        BOOLEAN DEFAULT FALSE,
    extnValue       OCTET STRING
                    -- 包含一个类型值的 DER 编码以及
                    -- 由 extnId 确定的扩展对象的 ExtnType。-- }

ExtensionSet EXTENSION      ::=  { ... }

EXTENSION ::= CLASS {
    &id       OBJECT IDENTIFIER UNIQUE,
    &ExtnType }
WITH SYNTAX {
    SYNTAX                  &ExtnType
    IDENTIFIED BY           &id }

-- 其他 PKI 证书构件

Certificates                ::=     SEQUENCE {
    userCertificate                 Certificate,
    certificationPath               ForwardCertificationPath OPTIONAL}

ForwardCertificationPath    ::=     SEQUENCE OF CrossCertificates

CrossCertificates           ::=     SET OF Certificate

CertificationPath           ::=     SEQUENCE {
    userCertificate                 Certificate,
    theCACertificates               SEQUENCE OF CertificatePair OPTIONAL}

CertificatePair         ::=     SEQUENCE {
    forward             [0]     Certificate OPTIONAL,
    reverse             [1]     Certificate OPTIONAL
                                -- 至少应出现一个对 -- }
    (WITH COMPONENTS { ..., forward PRESENT} |
    WITH COMPONENTS { ..., reverse PRESENT})

-- 证书撤消清单（CRL）

CertificateList         ::=     SIGNED { SEQUENCE {
    version                         Version OPTIONAL,
                            -- 如果出现，那么版本为 v2。
    signature                       AlgorithmIdentifier,
    issuer                          Name,
    thisUpdate                      Time,
    nextUpdate                      Time OPTIONAL,
    revokedCertificates             SEQUENCE OF SEQUENCE {
        serialNumber                    CertificateSerialNumber,
        revocationDate                  Time,
        crlEntryExtensions              Extensions OPTIONAL } OPTIONAL,
    crlExtensions       [0]         Extensions OPTIONAL }}

-- 信息对象类别 --

ALGORITHM   ::=   TYPE-IDENTIFIER

-- 参数化的类型 --
HASH {ToBeHashed}           ::=     SEQUENCE {
        algorithmIdentifier             AlgorithmIdentifier,
```

```
        hashValue                       BIT STRING ( CONSTRAINED BY {
            -- 将为把散列程序应用于 DER 编码的八比特组的结果 --

            -- 值为 -- ToBeHashed } ) }

ENCRYPTED-HASH { ToBeSigned }           ::=     BIT STRING ( CONSTRAINED BY {
    -- 将为把散列程序应用于 DER 编码（见第 6.1 节）的八比特组的结果 --
    -- 值为 -- ToBeSigned – 而后把加密程序应用于这些八比特组 -- })

ENCRYPTED { ToBeEnciphered }            ::=     BIT STRING ( CONSTRAINED BY {
    -- 将为把加密程序应用于 BER 编码的八比特组的结果 --
    -- 值为 -- ToBeEnciphered })


SIGNATURE { ToBeSigned }                ::=     SEQUENCE {
    algorithmIdentifier                 AlgorithmIdentifier,
    encrypted                           ENCRYPTED-HASH { ToBeSigned }}


SIGNED { ToBeSigned }                   ::=         SEQUENCE {
    toBeSigned                          ToBeSigned,
    COMPONENTS OF                       SIGNATURE { ToBeSigned }}


    -- PKI 对象类别 --

pkiUser  OBJECT-CLASS           ::= {
    SUBCLASS OF     {top}
    KIND            auxiliary
    MAY CONTAIN     {userCertificate}
    ID              id-oc-pkiUser }


pkiCA    OBJECT-CLASS           ::= {
    SUBCLASS OF     {top}
    KIND            auxiliary
    MAY CONTAIN     {cACertificate |
                    certificateRevocationList |
                    authorityRevocationList |
                    crossCertificatePair }
    ID              id-oc-pkiCA }


cRLDistributionPoint            OBJECT-CLASS            ::= {
    SUBCLASS OF                 { top }
    KIND                        structural
    MUST CONTAIN                { commonName }
    MAY CONTAIN                 { certificateRevocationList |
                                authorityRevocationList |
                                deltaRevocationList }
    ID                          id-oc-cRLDistributionPoint }


cRLDistPtNameForm  NAME-FORM        ::= {
    NAMES                       cRLDistributionPoint
    WITH ATTRIBUTES             { commonName}
    ID                          id-nf-cRLDistPtNameForm }


deltaCRL        OBJECT-CLASS        ::= {
    SUBCLASS OF             {top}
    KIND                    auxiliary
    MAY CONTAIN             {deltaRevocationList}
    ID                      id-oc-deltaCRL }


cpCps           OBJECT-CLASS        ::= {
    SUBCLASS OF             {top}
    KIND                    auxiliary
    MAY CONTAIN             {certificatePolicy |
                            certificationPracticeStmt}
    ID                      id-oc-cpCps }
```

```
pkiCertPath             OBJECT-CLASS        ::= {
    SUBCLASS OF             {top}
    KIND                    auxiliary
    MAY CONTAIN             { pkiPath }
    ID                      id-oc-pkiCertPath }


    -- PKI 号码簿属性 --

userCertificate                 ATTRIBUTE   ::=     {
    WITH SYNTAX                             Certificate
    EQUALITY MATCHING RULE                  certificateExactMatch
    ID                                      id-at-userCertificate}


cACertificate                   ATTRIBUTE   ::=     {
    WITH SYNTAX                             Certificate
    EQUALITY MATCHING RULE                  certificateExactMatch
    ID                                      id-at-cAcertificate }


crossCertificatePair            ATTRIBUTE   ::=     {
    WITH SYNTAX                             CertificatePair
    EQUALITY MATCHING RULE                  certificatePairExactMatch
    ID                                      id-at-crossCertificatePair }


certificateRevocationList ATTRIBUTE     ::=     {
    WITH SYNTAX                             CertificateList
    EQUALITY MATCHING RULE                  certificateListExactMatch
    ID                                      id-at-certificateRevocationList }


authorityRevocationList  ATTRIBUTE      ::=     {
    WITH SYNTAX                             CertificateList
    EQUALITY MATCHING RULE                  certificateListExactMatch
    ID                                      id-at-authorityRevocationList }


deltaRevocationList ATTRIBUTE ::= {
    WITH SYNTAX                             CertificateList
    EQUALITY MATCHING RULE                  certificateListExactMatch
    ID                                      id-at-deltaRevocationList }


supportedAlgorithms  ATTRIBUTE ::= {
    WITH SYNTAX                             SupportedAlgorithm
    EQUALITY MATCHING RULE                  algorithmIdentifierMatch
    ID                                      id-at-supportedAlgorithms }

SupportedAlgorithm ::= SEQUENCE {
    algorithmIdentifier                             AlgorithmIdentifier,
    intendedUsage                           [0]     KeyUsage OPTIONAL,
    intendedCertificatePolicies             [1]     CertificatePoliciesSyntax OPTIONAL }

certificationPracticeStmt ATTRIBUTE     ::=     {
    WITH SYNTAX         InfoSyntax
    ID                  id-at-certificationPracticeStmt }


InfoSyntax      ::=     CHOICE {
    content             DirectoryString {ub-content},
    pointer             SEQUENCE {
        name                    GeneralNames,
        hash                    HASH { HashedPolicyInfo } OPTIONAL } }


POLICY  ::=     TYPE-IDENTIFIER
```

```
HashedPolicyInfo       ::=     POLICY.&Type( {Policies} )

Policies POLICY ::= {...}  -- 由实施者定义 --


CertificatePolicy      ATTRIBUTE ::=        {
     WITH SYNTAX              PolicySyntax
     ID                       id-at-certificatePolicy }


PolicySyntax   ::=    SEQUENCE {
     policyIdentifier         PolicyID,
     policySyntax             InfoSyntax
     }


PolicyID            ::=        CertPolicyId


pkiPath        ATTRIBUTE      ::= {
     WITH SYNTAX              PkiPath
     ID                       id-at-pkiPath }


PkiPath        ::=       SEQUENCE OF Certificate


userPassword    ATTRIBUTE   ::=    {
     WITH SYNTAX                    OCTET STRING (SIZE (0..ub-user-password))
     EQUALITY MATCHING RULE         octetStringMatch
     ID                             id-at-userPassword }


-- 对象标识符赋值 --

-- 对象类别 --

id-oc-cRLDistributionPoint          OBJECT IDENTIFIER ::=          {id-oc 19}
id-oc-pkiUser                       OBJECT IDENTIFIER ::=          {id-oc 21}
id-oc-pkiCA                         OBJECT IDENTIFIER ::=          {id-oc 22}
id-oc-deltaCRL                      OBJECT IDENTIFIER ::=          {id-oc 23}
id-oc-cpCps                         OBJECT IDENTIFIER ::=          {id-oc 30}
id-oc-pkiCertPath                   OBJECT IDENTIFIER ::=          {id-oc 31}

-- 名称形式 --

id-nf-cRLDistPtNameForm             OBJECT IDENTIFIER ::=          {id-nf 14}

-- 号码簿属性 --

id-at-userPassword                  OBJECT IDENTIFIER ::=          {id-at 35}
id-at-userCertificate               OBJECT IDENTIFIER ::=          {id-at 36}
id-at-cAcertificate                 OBJECT IDENTIFIER ::=          {id-at 37}
id-at-authorityRevocationList       OBJECT IDENTIFIER ::=          {id-at 38}
id-at-certificateRevocationList     OBJECT IDENTIFIER ::=          {id-at 39}

id-at-crossCertificatePair          OBJECT IDENTIFIER ::=          {id-at 40}
id-at-supportedAlgorithms           OBJECT IDENTIFIER ::=          {id-at 52}
id-at-deltaRevocationList           OBJECT IDENTIFIER ::=          {id-at 53}
id-at-certificationPracticeStmt     OBJECT IDENTIFIER ::=          {id-at 68}
id-at-certificatePolicy             OBJECT IDENTIFIER ::=          {id-at 69}
id-at-pkiPath                       OBJECT IDENTIFIER ::=          {id-at 70}

END
```

-- *A.2* 证书扩展模块

```
CertificateExtensions {joint-iso-itu-t ds(5) module(1) certificateExtensions(26) 5}
DEFINITIONS IMPLICIT TAGS ::=
BEGIN
```

```
-- EXPORTS ALL --

IMPORTS
    id-at, id-ce, id-mr, informationFramework, authenticationFramework,
        selectedAttributeTypes, upperBounds
        FROM UsefulDefinitions {joint-iso-itu-t ds(5) module(1)
        usefulDefinitions(0) 5}

    Name, RelativeDistinguishedName, ATTRIBUTE, Attribute, MATCHING-RULE
        FROM InformationFramework informationFramework

    CertificateSerialNumber, CertificateList, AlgorithmIdentifier,
        EXTENSION, Time, PolicyID
        FROM AuthenticationFramework authenticationFramework

    DirectoryString {}
        FROM SelectedAttributeTypes selectedAttributeTypes

    ub-name
        FROM UpperBounds upperBounds

    ORAddress
        FROM MTSAbstractService {joint-iso-itu-t mhs(6) mts(3)
        modules(0) mts-abstract-service(1) version-1999 (1) } ;
-- 除非明确指出，否则
-- 对本规范中构件序列各部件的排序没有任何意义。

-- 公开密钥证书和CRL扩展 --


authorityKeyIdentifier EXTENSION ::= {
    SYNTAX              AuthorityKeyIdentifier
    IDENTIFIED BY       id-ce-authorityKeyIdentifier }

AuthorityKeyIdentifier ::= SEQUENCE {
    keyIdentifier               [0] KeyIdentifier               OPTIONAL,
    authorityCertIssuer         [1] GeneralNames                OPTIONAL,
    authorityCertSerialNumber   [2] CertificateSerialNumber     OPTIONAL }
    ( WITH COMPONENTS           {..., authorityCertIssuer PRESENT,
                    authorityCertSerialNumber PRESENT} |
    WITH COMPONENTS             {..., authorityCertIssuer ABSENT,
                    authorityCertSerialNumber ABSENT} )

KeyIdentifier ::= OCTET STRING


subjectKeyIdentifier EXTENSION ::= {
    SYNTAX              SubjectKeyIdentifier
    IDENTIFIED BY       id-ce-subjectKeyIdentifier }

SubjectKeyIdentifier ::= KeyIdentifier


keyUsage EXTENSION ::= {
      SYNTAX              KeyUsage
      IDENTIFIED BY       id-ce-keyUsage }

KeyUsage ::= BIT STRING {
    digitalSignature        (0),
    contentCommitment       (1),
    keyEncipherment         (2),
    dataEncipherment        (3),
    keyAgreement            (4),
    keyCertSign             (5),
    cRLSign                 (6),
    encipherOnly            (7),
    decipherOnly            (8) }

extKeyUsage EXTENSION ::= {
    SYNTAX              SEQUENCE SIZE (1..MAX) OF KeyPurposeId
    IDENTIFIED BY       id-ce-extKeyUsage }

KeyPurposeId ::= OBJECT IDENTIFIER
```

```
privateKeyUsagePeriod EXTENSION ::= {
    SYNTAX              PrivateKeyUsagePeriod
    IDENTIFIED BY       id-ce-privateKeyUsagePeriod }

PrivateKeyUsagePeriod ::= SEQUENCE {
    notBefore       [0]   GeneralizedTime OPTIONAL,
    notAfter        [1]   GeneralizedTime OPTIONAL }
    ( WITH COMPONENTS {..., notBefore PRESENT} |
    WITH COMPONENTS       {..., notAfter PRESENT} )


certificatePolicies EXTENSION ::= {
    SYNTAX              CertificatePoliciesSyntax
    IDENTIFIED BY       id-ce-certificatePolicies }

CertificatePoliciesSyntax ::= SEQUENCE SIZE (1..MAX) OF PolicyInformation

PolicyInformation ::= SEQUENCE {
    policyIdentifier        CertPolicyId,
    policyQualifiers        SEQUENCE SIZE (1..MAX) OF
                                PolicyQualifierInfo OPTIONAL }

CertPolicyId ::= OBJECT IDENTIFIER

PolicyQualifierInfo ::= SEQUENCE {
    policyQualifierId           CERT-POLICY-QUALIFIER.&id
                                ({SupportedPolicyQualifiers}),
    qualifier                   CERT-POLICY-QUALIFIER.&Qualifier
                                ({SupportedPolicyQualifiers}{@policyQualifierId})
                                OPTIONAL }

SupportedPolicyQualifiers CERT-POLICY-QUALIFIER ::= { ... }

anyPolicy       OBJECT IDENTIFIER       ::=         { 2 5 29 32 0 }

CERT-POLICY-QUALIFIER ::= CLASS {
    &id                 OBJECT IDENTIFIER UNIQUE,
    &Qualifier          OPTIONAL }
WITH SYNTAX {
    POLICY-QUALIFIER-ID       &id
    [QUALIFIER-TYPE       &Qualifier] }


policyMappings EXTENSION ::= {
    SYNTAX              PolicyMappingsSyntax
    IDENTIFIED BY       id-ce-policyMappings }

PolicyMappingsSyntax ::= SEQUENCE SIZE (1..MAX) OF SEQUENCE {
    issuerDomainPolicy          CertPolicyId,
    subjectDomainPolicy         CertPolicyId }


subjectAltName EXTENSION ::= {
    SYNTAX              GeneralNames
    IDENTIFIED BY       id-ce-subjectAltName }

GeneralNames ::= SEQUENCE SIZE (1..MAX) OF GeneralName

GeneralName ::= CHOICE {
    otherName                   [0]     INSTANCE OF OTHER-NAME,
    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 }

OTHER-NAME ::= TYPE-IDENTIFIER

EDIPartyName ::= SEQUENCE {
    nameAssigner        [0]     DirectoryString {ub-name} OPTIONAL,
    partyName           [1]     DirectoryString {ub-name} }
```

```
issuerAltName EXTENSION ::= {
    SYNTAX              GeneralNames
    IDENTIFIED BY       id-ce-issuerAltName }


subjectDirectoryAttributes EXTENSION ::= {
    SYNTAX              AttributesSyntax
    IDENTIFIED BY       id-ce-subjectDirectoryAttributes }

AttributesSyntax ::= SEQUENCE SIZE (1..MAX) OF Attribute


basicConstraints EXTENSION ::= {
    SYNTAX              BasicConstraintsSyntax
    IDENTIFIED BY       id-ce-basicConstraints }

BasicConstraintsSyntax ::= SEQUENCE {
    cA                      BOOLEAN DEFAULT FALSE,
    pathLenConstraint       INTEGER (0..MAX) OPTIONAL }


nameConstraints EXTENSION ::= {
    SYNTAX        NameConstraintsSyntax
    IDENTIFIED BY       id-ce-nameConstraint }

NameConstraintsSyntax ::= SEQUENCE {
    permittedSubtrees       [0]    GeneralSubtrees OPTIONAL,
    excludedSubtrees        [1]    GeneralSubtrees OPTIONAL,
    requiredNameForms       [2]    NameForms OPTIONAL  }

GeneralSubtrees ::= SEQUENCE SIZE (1..MAX) OF GeneralSubtree

GeneralSubtree ::= SEQUENCE {
    base                    GeneralName,
    minimum     [0]    BaseDistance DEFAULT 0,
    maximum     [1]    BaseDistance OPTIONAL }

BaseDistance ::= INTEGER (0..MAX)


NameForms  ::= SEQUENCE {
   basicNameForms       [0]        BasicNameForms OPTIONAL,
   otherNameForms       [1]        SEQUENCE SIZE (1..MAX) OF OBJECT IDENTIFIER OPTIONAL }
(ALL EXCEPT ({ -- 无；即至少应出现一个部件。 -- }))


BasicNameForms  ::= BIT STRING {
          rfc822Name            (0),
    dNSName             (1),
    x400Address         (2),
    directoryName       (3),
    ediPartyName        (4),
    uniformResourceIdentifier    (5),
    iPAddress           (6),
    registeredID                (7) }  (SIZE (1..MAX))


policyConstraints EXTENSION ::= {
    SYNTAX              PolicyConstraintsSyntax
    IDENTIFIED BY       id-ce-policyConstraints }

PolicyConstraintsSyntax ::= SEQUENCE {
    requireExplicitPolicy    [0] SkipCerts OPTIONAL,
    inhibitPolicyMapping     [1] SkipCerts OPTIONAL }

SkipCerts ::= INTEGER (0..MAX)


cRLNumber EXTENSION ::= {
    SYNTAX              CRLNumber
    IDENTIFIED BY       id-ce-cRLNumber }

CRLNumber ::= INTEGER (0..MAX)
```

```
reasonCode EXTENSION ::= {
    SYNTAX              CRLReason
    IDENTIFIED BY       id-ce-reasonCode }

CRLReason ::= ENUMERATED {
    unspecified                 (0),
    keyCompromise               (1),
    cACompromise                (2),
    affiliationChanged          (3),
    superseded                  (4),
    cessationOfOperation        (5),
    certificateHold             (6),
    removeFromCRL               (8),
    privilegeWithdrawn          (9),
    aaCompromise                (10) }


holdInstructionCode EXTENSION ::= {
    SYNTAX              HoldInstruction
    IDENTIFIED BY       id-ce-instructionCode }

HoldInstruction ::= OBJECT IDENTIFIER


invalidityDate EXTENSION ::= {
    SYNTAX              GeneralizedTime
    IDENTIFIED BY       id-ce-invalidityDate }


crlScope EXTENSION ::= {
    SYNTAX                CRLScopeSyntax
    IDENTIFIED BY         id-ce-cRLScope }

CRLScopeSyntax ::=        SEQUENCE SIZE (1..MAX) OF PerAuthorityScope

PerAuthorityScope ::= SEQUENCE {
    authorityName               [0]     GeneralName OPTIONAL,
    distributionPoint           [1]     DistributionPointName OPTIONAL,
    onlyContains                [2]     OnlyCertificateTypes OPTIONAL,
    onlySomeReasons             [4]     ReasonFlags OPTIONAL,
    serialNumberRange           [5]     NumberRange OPTIONAL,
    subjectKeyIdRange           [6]     NumberRange OPTIONAL,

    nameSubtrees                [7]     GeneralNames OPTIONAL,
    baseRevocationInfo          [9]     BaseRevocationInfo OPTIONAL
    }

OnlyCertificateTypes ::= BIT STRING {
    user        (0),
    authority   (1),
    attribute   (2) }

NumberRange ::= SEQUENCE {
    startingNumber      [0]     INTEGER OPTIONAL,
    endingNumber        [1]     INTEGER OPTIONAL,
    modulus                     INTEGER OPTIONAL }

BaseRevocationInfo ::= SEQUENCE {
    cRLStreamIdentifier         [0]     CRLStreamIdentifier     OPTIONAL,
    cRLNumber                   [1]     CRLNumber,
    baseThisUpdate              [2]     GeneralizedTime }


statusReferrals EXTENSION  ::= {
    SYNTAX              StatusReferrals
    IDENTIFIED BY       id-ce-statusReferrals }

StatusReferrals ::= SEQUENCE SIZE (1..MAX) OF StatusReferral

StatusReferral  ::=  CHOICE {
    cRLReferral         [0]     CRLReferral,
    otherReferral       [1]     INSTANCE OF OTHER-REFERRAL}

CRLReferral ::= SEQUENCE  {
    issuer              [0]     GeneralName OPTIONAL,
    location            [1]     GeneralName OPTIONAL,
    deltaRefInfo        [2]     DeltaRefInfo OPTIONAL,
    cRLScope                    CRLScopeSyntax,
```

```
    lastUpdate          [3]     GeneralizedTime OPTIONAL,
    lastChangedCRL      [4]     GeneralizedTime OPTIONAL}

DeltaRefInfo ::= SEQUENCE {
    deltaLocation       GeneralName,
    lastDelta           GeneralizedTime OPTIONAL }

OTHER-REFERRAL ::= TYPE-IDENTIFIER


cRLStreamIdentifier EXTENSION ::= {
    SYNTAX              CRLStreamIdentifier
    IDENTIFIED BY       id-ce-cRLStreamIdentifier }


CRLStreamIdentifier ::= INTEGER (0..MAX)


orderedList EXTENSION ::= {
    SYNTAX              OrderedListSyntax
    IDENTIFIED BY       id-ce-orderedList }

OrderedListSyntax ::= ENUMERATED {
ascSerialNum        (0),
ascRevDate          (1) }


deltaInfo EXTENSION     ::=     {
    SYNTAX              DeltaInformation
    IDENTIFIED BY       id-ce-deltaInfo }

DeltaInformation        ::=     SEQUENCE {
    deltaLocation       GeneralName,
    nextDelta           GeneralizedTime OPTIONAL }


cRLDistributionPoints EXTENSION ::= {
    SYNTAX              CRLDistPointsSyntax
    IDENTIFIED BY       id-ce-cRLDistributionPoints }

CRLDistPointsSyntax ::= 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 }

ReasonFlags ::= BIT STRING {
    unused                  (0),
    keyCompromise           (1),
    cACompromise            (2),
    affiliationChanged      (3),
    superseded              (4),
    cessationOfOperation    (5),
    certificateHold         (6),
    privilegeWithdrawn      (7),
    aACompromise            (8) }


issuingDistributionPoint EXTENSION  ::= {
    SYNTAX IssuingDistPointSyntax
    IDENTIFIED BY id-ce-issuingDistributionPoint }

IssuingDistPointSyntax ::= SEQUENCE {
    -- 如果 onlyContainsUserPublicKeyCerts 和 onlyContainsCACerts 都为 FALSE,
    -- 那么 CRL 涵盖两种证书类型。
    distributionPoint                   [0] DistributionPointName OPTIONAL,
    onlyContainsUserPublicKeyCerts      [1] BOOLEAN DEFAULT FALSE,
    onlyContainsCACerts                 [2] BOOLEAN DEFAULT FALSE,
```

```
    onlySomeReasons                 [3] ReasonFlags OPTIONAL,
    indirectCRL                     [4] BOOLEAN DEFAULT FALSE }

certificateIssuer EXTENSION ::= {
    SYNTAX              GeneralNames
    IDENTIFIED BY       id-ce-certificateIssuer }

deltaCRLIndicator EXTENSION ::= {
    SYNTAX              BaseCRLNumber
    IDENTIFIED BY       id-ce-deltaCRLIndicator }

BaseCRLNumber ::= CRLNumber

toBeRevoked  EXTENSION  ::= {
    SYNTAX      ToBeRevokedSyntax
    IDENTIFIED BY   id-ce-toBeRevoked }

ToBeRevokedSyntax::=  SEQUENCE SIZE(1..MAX) OF ToBeRevokedGroup

ToBeRevokedGroup      ::=       SEQUENCE {
    certificateIssuer     [0]       GeneralName OPTIONAL,
    reasonInfo            [1]       ReasonInfo      OPTIONAL,
    revocationTime                  GeneralizedTime,
    certificateGroup                CertificateGroup }

ReasonInfo      ::=       SEQUENCE {
    reasonCode                      CRLReason,
    holdInstructionCode HoldInstruction OPTIONAL }

CertificateGroup::=       CHOICE {
    serialNumbers         [0]       CertificateSerialNumbers,
    serialNumberRange  [1]       CertificateGroupNumberRange,

    nameSubtree           [2]       GeneralName }

CertificateGroupNumberRange ::= SEQUENCE {
    startingNumber        [0]       INTEGER,
    endingNumber          [1]       INTEGER }

CertificateSerialNumbers ::=      SEQUENCE SIZE(1..MAX) OF CertificateSerialNumber

revokedGroups  EXTENSION  ::= {
    SYNTAX      RevokedGroupsSyntax
    IDENTIFIED BY   id-ce-RevokedGroups }

RevokedGroupsSyntax ::=           SEQUENCE SIZE (1..MAX) OF RevokedGroup

RevokedGroup ::=    SEQUENCE {
    certificateIssuer               [0]       GeneralName OPTIONAL,
    reasonInfo                      [1]       ReasonInfo OPTIONAL,
    invalidityDate                  [2]       GeneralizedTime OPTIONAL,
    revokedcertificateGroup         [3]       RevokedCertificateGroup }

RevokedCertificateGroup ::=       CHOICE {
    serialNumberRange  NumberRange,
    nameSubtree           GeneralName }

expiredCertsOnCRL EXTENSION  ::= {
    SYNTAX ExpiredCertsOnCRL
    IDENTIFIED BY id-ce-expiredCertsOnCRL }

ExpiredCertsOnCRL       ::= GeneralizedTime

baseUpdateTime EXTENSION      ::= {
    SYNTAX              GeneralizedTime
    IDENTIFIED BY       id-ce-baseUpdateTime }


freshestCRL EXTENSION      ::= {
    SYNTAX              CRLDistPointsSyntax
    IDENTIFIED BY       id-ce-freshestCRL }
```

```
aAissuingDistributionPoint EXTENSION ::= {
    SYNTAX AAIssuingDistPointSyntax
    IDENTIFIED BY id-ce-aAissuingDistributionPoint }

AAIssuingDistPointSyntax ::= SEQUENCE {
    distributionPoint           [ 0 ] DistributionPointName OPTIONAL,
    onlySomeReasons             [ 1 ] ReasonFlags OPTIONAL,
    indirectCRL                 [ 2 ] BOOLEAN DEFAULT FALSE,
    containsUserAttributeCerts  [ 3 ] BOOLEAN DEFAULT TRUE,
    containsAACerts             [ 4 ] BOOLEAN DEFAULT TRUE,
    containsSOAPublicKeyCerts   [ 5 ] BOOLEAN DEFAULT TRUE }


inhibitAnyPolicy EXTENSION      ::= {
    SYNTAX      SkipCerts
IDENTIFIED BY   id-ce-inhibitAnyPolicy }
```

-- PKI 匹配规则 --

```
certificateExactMatch MATCHING-RULE ::= {
    SYNTAX      CertificateExactAssertion
    ID          id-mr-certificateExactMatch }

CertificateExactAssertion ::= SEQUENCE {
    serialNumber      CertificateSerialNumber,
    issuer            Name }

certificateMatch MATCHING-RULE ::= {
    SYNTAX      CertificateAssertion
    ID          id-mr-certificateMatch }

CertificateAssertion ::= SEQUENCE {
    serialNumber            [0] CertificateSerialNumber         OPTIONAL,
    issuer                  [1] Name                            OPTIONAL,
    subjectKeyIdentifier    [2] SubjectKeyIdentifier            OPTIONAL,
    authorityKeyIdentifier   [3] AuthorityKeyIdentifier         OPTIONAL,
    certificateValid        [4] Time                            OPTIONAL,
    privateKeyValid         [5] GeneralizedTime                 OPTIONAL,
    subjectPublicKeyAlgID   [6] OBJECT IDENTIFIER               OPTIONAL,
    keyUsage                [7] KeyUsage                        OPTIONAL,
    subjectAltName          [8] AltNameType                     OPTIONAL,
    policy                  [9] CertPolicySet                   OPTIONAL,
    pathToName              [10] Name                           OPTIONAL,
    subject                 [11]  Name                          OPTIONAL,
    nameConstraints         [12]  NameConstraintsSyntax         OPTIONAL }

AltNameType ::= CHOICE {
    builtinNameForm     ENUMERATED {
        rfc822Name                  (1),
        dNSName                     (2),
        x400Address                 (3),
        directoryName               (4),
        ediPartyName                (5),
        uniformResourceIdentifier   (6),
        iPAddress                   (7),
        registeredId                (8) },
    otherNameForm       OBJECT IDENTIFIER }

CertPolicySet ::= SEQUENCE SIZE (1..MAX) OF CertPolicyId


certificatePairExactMatch MATCHING-RULE         ::=         {
    SYNTAX          CertificatePairExactAssertion
    ID              id-mr-certificatePairExactMatch }

CertificatePairExactAssertion ::= SEQUENCE {
    issuedToThisCAAssertion     [0] CertificateExactAssertion OPTIONAL,
    issuedByThisCAAssertion     [1] CertificateExactAssertion OPTIONAL }
    ( WITH COMPONENTS           {..., issuedToThisCAAssertion PRESENT} |
      WITH COMPONENTS           {..., issuedByThisCAAssertion PRESENT} )
```

```
certificatePairMatch MATCHING-RULE ::= {
    SYNTAX        CertificatePairAssertion
    ID            id-mr-certificatePairMatch }

CertificatePairAssertion ::= SEQUENCE {
    issuedToThisCAAssertion    [0] CertificateAssertion OPTIONAL,
    issuedByThisCAAssertion    [1] CertificateAssertion OPTIONAL }
    ( WITH COMPONENTS      {..., issuedToThisCAAssertion PRESENT} |
      WITH COMPONENTS      {..., issuedByThisCAAssertion PRESENT} )


certificateListExactMatch MATCHING-RULE        ::=       {
    SYNTAX          CertificateListExactAssertion
    ID              id-mr-certificateListExactMatch }

CertificateListExactAssertion ::= SEQUENCE {
    issuer              Name,
    thisUpdate          Time,
    distributionPoint   DistributionPointName OPTIONAL }


certificateListMatch MATCHING-RULE ::= {
    SYNTAX          CertificateListAssertion
    ID              id-mr-certificateListMatch }

CertificateListAssertion ::= SEQUENCE {
    issuer                          Name                OPTIONAL,
    minCRLNumber            [0]     CRLNumber           OPTIONAL,
    maxCRLNumber            [1]     CRLNumber           OPTIONAL,
    reasonFlags                     ReasonFlags         OPTIONAL,
    dateAndTime                     Time                OPTIONAL,
    distributionPoint       [2]     DistributionPointName OPTIONAL,
    authorityKeyIdentifier  [3]     AuthorityKeyIdentifier OPTIONAL}


algorithmIdentifierMatch MATCHING-RULE         ::=   {
    SYNTAX          AlgorithmIdentifier
    ID              id-mr-algorithmIdentifierMatch }


policyMatch MATCHING-RULE    ::= {
    SYNTAX          PolicyID
    ID              id-mr-policyMatch }


pkiPathMatch MATCHING-RULE  ::= {
SYNTAX          PkiPathMatchSyntax
ID              id-mr-pkiPathMatch }


PkiPathMatchSyntax      ::= SEQUENCE {
firstIssuer     Name,
lastSubject     Name }


enhancedCertificateMatch MATCHING-RULE  ::= {
    SYNTAX      EnhancedCertificateAssertion
    ID          id-mr-enhancedCertificateMatch }

EnhancedCertificateAssertion  ::= SEQUENCE {
    serialNumber            [0]  CertificateSerialNumber  OPTIONAL,
    issuer                  [1]  Name                     OPTIONAL,

    subjectKeyIdentifier    [2]  SubjectKeyIdentifier     OPTIONAL,
    authorityKeyIdentifier  [3]  AuthorityKeyIdentifier   OPTIONAL,
    certificateValid        [4]  Time                     OPTIONAL,
    privateKeyValid         [5]  GeneralizedTime          OPTIONAL,
    subjectPublicKeyAlgID   [6]  OBJECT IDENTIFIER        OPTIONAL,
    keyUsage                [7]  KeyUsage                 OPTIONAL,
    subjectAltName          [8]  AltName                  OPTIONAL,
    policy                  [9]  CertPolicySet            OPTIONAL,
    pathToName              [10] GeneralNames             OPTIONAL,
    subject                 [11] Name                     OPTIONAL,
    nameConstraints         [12] NameConstraintsSyntax    OPTIONAL
}
```

```
(ALL EXCEPT ({ -- 无；至少应出现一个部件。 -- }))

AltName  ::= SEQUENCE {
        altnameType         AltNameType,
        altNameValue        GeneralName OPTIONAL }


-- 对象标识符赋值 --

id-ce-subjectDirectoryAttributes        OBJECT IDENTIFIER   ::=   {id-ce 9}
id-ce-subjectKeyIdentifier              OBJECT IDENTIFIER   ::=   {id-ce 14}
id-ce-keyUsage                          OBJECT IDENTIFIER   ::=   {id-ce 15}
id-ce-privateKeyUsagePeriod             OBJECT IDENTIFIER   ::=   {id-ce 16}
id-ce-subjectAltName                    OBJECT IDENTIFIER   ::=   {id-ce 17}
id-ce-issuerAltName                     OBJECT IDENTIFIER   ::=   {id-ce 18}
id-ce-basicConstraints                  OBJECT IDENTIFIER   ::=   {id-ce 19}
id-ce-cRLNumber                         OBJECT IDENTIFIER   ::=   {id-ce 20}
id-ce-reasonCode                        OBJECT IDENTIFIER   ::=   {id-ce 21}
id-ce-instructionCode                   OBJECT IDENTIFIER   ::=   {id-ce 23}
id-ce-invalidityDate                    OBJECT IDENTIFIER   ::=   {id-ce 24}
id-ce-deltaCRLIndicator                 OBJECT IDENTIFIER   ::=   {id-ce 27}
id-ce-issuingDistributionPoint          OBJECT IDENTIFIER   ::=   {id-ce 28}

id-ce-certificateIssuer                 OBJECT IDENTIFIER   ::=   {id-ce 29}
id-ce-nameConstraint                    OBJECT IDENTIFIER   ::=   {id-ce 30 1}


id-ce-cRLDistributionPoints             OBJECT IDENTIFIER   ::=   {id-ce 31}
id-ce-certificatePolicies               OBJECT IDENTIFIER   ::=   {id-ce 32}
id-ce-policyMappings                    OBJECT IDENTIFIER   ::=   {id-ce 33}
-- 不赞成                               OBJECT IDENTIFIER   ::=   {id-ce 34}
id-ce-authorityKeyIdentifier            OBJECT IDENTIFIER   ::=   {id-ce 35}
id-ce-policyConstraints                 OBJECT IDENTIFIER   ::=   {id-ce 36}
id-ce-extKeyUsage                       OBJECT IDENTIFIER   ::=   {id-ce 37}
id-ce-cRLStreamIdentifier               OBJECT IDENTIFIER   ::=   {id-ce 40}
id-ce-cRLScope                          OBJECT IDENTIFIER   ::=   {id-ce 44}
id-ce-statusReferrals                   OBJECT IDENTIFIER   ::=   {id-ce 45}
id-ce-freshestCRL                       OBJECT IDENTIFIER   ::=   {id-ce 46}
id-ce-orderedList                       OBJECT IDENTIFIER   ::=   {id-ce 47}
id-ce-baseUpdateTime                    OBJECT IDENTIFIER   ::=   {id-ce 51}
id-ce-deltaInfo                         OBJECT IDENTIFIER   ::=   {id-ce 53}
id-ce-inhibitAnyPolicy                  OBJECT IDENTIFIER   ::=   {id-ce 54}
id-ce-toBeRevoked                       OBJECT IDENTIFIER   ::=   {id-ce 58}
id-ce-RevokedGroups                     OBJECT IDENTIFIER   ::=   {id-ce 59}
id-ce-expiredCertsOnCRL                 OBJECT IDENTIFIER   ::=   {id-ce 60}
id-ce-aAissuingDistributionPoint        OBJECT IDENTIFIER   ::=   {id-ce 63}


-- 匹配规则 OID --

id-mr-certificateExactMatch             OBJECT IDENTIFIER   ::=   {id-mr 34}
id-mr-certificateMatch                  OBJECT IDENTIFIER   ::=   {id-mr 35}
id-mr-certificatePairExactMatch         OBJECT IDENTIFIER   ::=   {id-mr 36}
id-mr-certificatePairMatch              OBJECT IDENTIFIER   ::=   {id-mr 37}
id-mr-certificateListExactMatch         OBJECT IDENTIFIER   ::=   {id-mr 38}
id-mr-certificateListMatch              OBJECT IDENTIFIER   ::=   {id-mr 39}
id-mr-algorithmIdentifierMatch          OBJECT IDENTIFIER   ::=   {id-mr 40}
id-mr-policyMatch                       OBJECT IDENTIFIER   ::=   {id-mr 60}
id-mr-pkiPathMatch                      OBJECT IDENTIFIER   ::=   {id-mr 62}
id-mr-enhancedCertificateMatch          OBJECT IDENTIFIER   ::=   {id-mr 65}


-- 本规范不使用以下对象标识符（OBJECT IDENTIFIERS）：
-- {id-ce 2}、 {id-ce 3}、 {id-ce 4}、 {id-ce 5}、 {id-ce 6}、 {id-ce 7}、
-- {id-ce 8}、 {id-ce 10}、 {id-ce 11}、 {id-ce 12}、 {id-ce 13}、
-- {id-ce 22}、 {id-ce 25}、 {id-ce 26}、 {id-ce 30}

END
```

-- *A.3* 属性证书框架模块

```
AttributeCertificateDefinitions {joint-iso-itu-t ds(5) module(1) attributeCertificateDefinitions(32) 5}
DEFINITIONS IMPLICIT TAGS ::=
BEGIN

-- EXPORTS ALL --

IMPORTS
    id-at, id-ce, id-mr, informationFramework, authenticationFramework,
        selectedAttributeTypes, upperBounds, id-oc, certificateExtensions
        FROM UsefulDefinitions {joint-iso-itu-t ds(5) module(1)
        usefulDefinitions(0) 5}


    Name, RelativeDistinguishedName, ATTRIBUTE, Attribute,
        MATCHING-RULE, AttributeType, OBJECT-CLASS, top
        FROM InformationFramework informationFramework

    CertificateSerialNumber, CertificateList, AlgorithmIdentifier,
        EXTENSION, SIGNED{}, InfoSyntax, PolicySyntax, Extensions, Certificate
        FROM AuthenticationFramework authenticationFramework

    DirectoryString{}, TimeSpecification, UniqueIdentifier
        FROM SelectedAttributeTypes selectedAttributeTypes

    GeneralName, GeneralNames, NameConstraintsSyntax, certificateListExactMatch
        FROM CertificateExtensions certificateExtensions

    ub-name
        FROM UpperBounds upperBounds

    UserNotice
        FROM PKIX1Implicit93 {iso(1) identified-organization(3) dod(6) internet(1) security(5) mechanisms(5) pkix(7) id-
mod(0) id-pkix1-implicit-93(4)}

    ORAddress
        FROM MTSAbstractService {joint-iso-itu-t mhs(6) mts(3)
        modules(0) mts-abstract-service(1) version-1999 (1) } ;
-- 除非明确指出，否则
-- 对本规范中构件序列各部件的排序没有任何意义。

-- 属性证书构件 --

AttributeCertificate ::= SIGNED {AttributeCertificateInfo}

AttributeCertificateInfo ::= SEQUENCE
    {
    version                 AttCertVersion, -- 版本为 v2
    holder                  Holder,
    issuer                  AttCertIssuer,
    signature               AlgorithmIdentifier,
    serialNumber            CertificateSerialNumber,
    attrCertValidityPeriod  AttCertValidityPeriod,
    attributes              SEQUENCE OF Attribute,
    issuerUniqueID          UniqueIdentifier OPTIONAL,
    extensions              Extensions              OPTIONAL
    }

AttCertVersion   ::= INTEGER { v2(1) }

Holder   ::=      SEQUENCE
        {
        baseCertificateID       [0] IssuerSerial        OPTIONAL,
            -- 持有者公开密钥证书的发布者和序列号
        entityName              [1] GeneralNames        OPTIONAL,
            -- 实体或角色的名称
        objectDigestInfo        [2] ObjectDigestInfo    OPTIONAL
            -- 用于直接鉴别持有者，例如，一个可执行者
-- baseCertificateID、entityName 或 objectDigestInfo 中至少应出现一个 --}
```

```
ObjectDigestInfo   ::= SEQUENCE {
     digestedObjectType  ENUMERATED {
          publicKey             (0),
          publicKeyCert         (1),
          otherObjectTypes      (2) },
     otherObjectTypeID         OBJECT IDENTIFIER  OPTIONAL,
     digestAlgorithm           AlgorithmIdentifier,
     objectDigest              BIT STRING }


AttCertIssuer     ::=    [0] SEQUENCE {
     issuerName                     GeneralNames  OPTIONAL,
     baseCertificateID      [0]     IssuerSerial  OPTIONAL,
     objectDigestInfo       [1]     ObjectDigestInfo  OPTIONAL }


-- 至少应出现一个部件。
     ( WITH COMPONENTS { ..., issuerName  PRESENT } |
       WITH COMPONENTS { ..., baseCertificateID  PRESENT } |
       WITH COMPONENTS { ..., objectDigestInfo PRESENT } )


IssuerSerial  ::=  SEQUENCE {
     issuer           GeneralNames,
     serial           CertificateSerialNumber,
     issuerUID        UniqueIdentifier OPTIONAL }

AttCertValidityPeriod  ::= SEQUENCE {
     notBeforeTime      GeneralizedTime,
     notAfterTime       GeneralizedTime }

AttributeCertificationPath  ::=  SEQUENCE {
     attributeCertificate          AttributeCertificate,
     acPath                        SEQUENCE OF ACPathData OPTIONAL }

ACPathData  ::=  SEQUENCE {
     certificate             [0]  Certificate  OPTIONAL,
     attributeCertificate    [1]  AttributeCertificate  OPTIONAL }


PrivilegePolicy   ::=       OBJECT IDENTIFIER


-- 特权属性 --


role       ATTRIBUTE      ::= {
           WITH SYNTAX           RoleSyntax
           ID                    id-at-role }


xmlPrivilegeInfo       ATTRIBUTE  ::= {
     WITH SYNTAX   UTF8String --contains XML-encoded privilege information
     ID            id-at-xMLPrivilegeInfo }


RoleSyntax       ::=    SEQUENCE {
     roleAuthority          [0]     GeneralNames OPTIONAL,
     roleName               [1]     GeneralName }


-- PMI 对象类别 --


pmiUser OBJECT-CLASS ::= {
     SUBCLASS OF       {top}
     KIND              auxiliary
     MAY CONTAIN       {attributeCertificateAttribute}
     ID                id-oc-pmiUser
     }


pmiAA OBJECT-CLASS ::= {
-- 一个 PMI AA
     SUBCLASS OF       {top}
     KIND              auxiliary
```

```
    MAY CONTAIN     {aACertificate |
                    attributeCertificateRevocationList |
                    attributeAuthorityRevocationList}
    ID              id-oc-pmiAA
    }

pmiSOA OBJECT-CLASS ::= {  -- 一个 PMI 机构源
    SUBCLASS OF     {top}
    KIND            auxiliary
    MAY CONTAIN     {attributeCertificateRevocationList |
                    attributeAuthorityRevocationList |
                    attributeDescriptorCertificate}
    ID              id-oc-pmiSOA
    }

attCertCRLDistributionPt        OBJECT-CLASS ::= {
    SUBCLASS OF                 {top}
    KIND                        auxiliary
    MAY CONTAIN                 { attributeCertificateRevocationList |
                                attributeAuthorityRevocationList }
    ID                          id-oc-attCertCRLDistributionPts
    }

pmiDelegationPath               OBJECT-CLASS ::= {
    SUBCLASS OF                 {top}
    KIND                        auxiliary
    MAY CONTAIN                 { delegationPath }
    ID                          id-oc-pmiDelegationPath }

privilegePolicy                 OBJECT-CLASS ::= {
    SUBCLASS OF                 {top}
    KIND                        auxiliary
    MAY CONTAIN                 {privPolicy }
    ID                          id-oc-privilegePolicy }

protectedPrivilegePolicy        OBJECT-CLASS ::= {
    SUBCLASS OF     {top}
    KIND            auxiliary
    MAY CONTAIN     {protPrivPolicy }
    ID              id-oc-protectedPrivilegePolicy }

-- PMI 号码簿属性 --

attributeCertificateAttribute           ATTRIBUTE  ::= {
    WITH SYNTAX                         AttributeCertificate
    EQUALITY MATCHING RULE              attributeCertificateExactMatch
    ID                                  id-at-attributeCertificate }

aACertificate                           ATTRIBUTE      ::=      {
    WITH SYNTAX                         AttributeCertificate
    EQUALITY MATCHING RULE              attributeCertificateExactMatch
    ID                                  id-at-aACertificate }

attributeDescriptorCertificate          ATTRIBUTE  ::= {
    WITH SYNTAX                         AttributeCertificate
    EQUALITY MATCHING RULE              attributeCertificateExactMatch
    ID                                  id-at-attributeDescriptorCertificate }

attributeCertificateRevocationList      ATTRIBUTE ::= {
    WITH SYNTAX                         CertificateList
    EQUALITY MATCHING RULE              certificateListExactMatch
    ID                                  id-at-attributeCertificateRevocationList}

attributeAuthorityRevocationList        ATTRIBUTE      ::= {
    WITH SYNTAX                         CertificateList
    EQUALITY MATCHING RULE              certificateListExactMatch
    ID                                  id-at-attributeAuthorityRevocationList }
```

```
delegationPath              ATTRIBUTE       ::= {
    WITH SYNTAX             AttCertPath
    ID                      id-at-delegationPath }

AttCertPath     ::=     SEQUENCE OF AttributeCertificate

privPolicy                  ATTRIBUTE       ::= {
    WITH SYNTAX             PolicySyntax
    ID                      id-at-privPolicy }

protPrivPolicy  ATTRIBUTE       ::= {
    WITH SYNTAX                     AttributeCertificate
    EQUALITY MATCHING RULE          attributeCertificateExactMatch
    ID                              id-at-protPrivPolicy }

xmlPrivPolicy ATTRIBUTE         ::= {
    WITH SYNTAX     UTF8String -- 包含 XML 编码的特权策略信息
    ID          id-at-xMLPprotPrivPolicy }


-- 属性证书扩展和匹配规则 --


attributeCertificateExactMatch MATCHING-RULE  ::= {
    SYNTAX      AttributeCertificateExactAssertion
    ID          id-mr-attributeCertificateExactMatch }


AttributeCertificateExactAssertion      ::= SEQUENCE {
    serialNumber        CertificateSerialNumber,
    issuer              AttCertIssuer
    }


attributeCertificateMatch  MATCHING-RULE  ::=  {
    SYNTAX          AttributeCertificateAssertion
    ID              id-mr-attributeCertificateMatch  }


AttributeCertificateAssertion  ::=  SEQUENCE  {
    holder              [0]     CHOICE {
        baseCertificateID       [0]  IssuerSerial,
        holderName              [1]  GeneralNames} OPTIONAL,
    issuer              [1]     GeneralNames OPTIONAL,
    attCertValidity     [2]     GeneralizedTime OPTIONAL,
    attType             [3]     SET OF AttributeType OPTIONAL}
-- 至少应出现一个序列的部件。


holderIssuerMatch MATCHING-RULE  ::= {
    SYNTAX      HolderIssuerAssertion
    ID          id-mr-holderIssuerMatch }

HolderIssuerAssertion   ::=     SEQUENCE {
    holder      [0]             Holder   OPTIONAL,
    issuer      [1]             AttCertIssuer     OPTIONAL
    }


delegationPathMatch MATCHING-RULE  ::= {
    SYNTAX      DelMatchSyntax
    ID          id-mr-delegationPathMatch }


DelMatchSyntax ::=  SEQUENCE {
    firstIssuer   AttCertIssuer,
    lastHolder    Holder }


sOAIdentifier EXTENSION  ::= {
    SYNTAX              NULL
    IDENTIFIED BY       id-ce-sOAIdentifier }
```

```
sOAIdentifierMatch MATCHING-RULE ::= {
    SYNTAX              NULL
    ID                  id-mr-sOAIdentifierMatch }

authorityAttributeIdentifier EXTENSION  ::=
    {
    SYNTAX              AuthorityAttributeIdentifierSyntax
    IDENTIFIED BY       { id-ce-authorityAttributeIdentifier } }

AuthorityAttributeIdentifierSyntax          ::=     SEQUENCE SIZE (1..MAX) OF AuthAttId

AuthAttId       ::=     IssuerSerial

authAttIdMatch MATCHING-RULE  ::= {
    SYNTAX      AuthorityAttributeIdentifierSyntax
    ID          id-mr-authAttIdMatch }

roleSpecCertIdentifier EXTENSION ::=
    {
    SYNTAX              RoleSpecCertIdentifierSyntax
    IDENTIFIED BY       { id-ce-roleSpecCertIdentifier } }

RoleSpecCertIdentifierSyntax  ::= SEQUENCE SIZE (1..MAX) OF RoleSpecCertIdentifier

RoleSpecCertIdentifier  ::= SEQUENCE {
    roleName                [0]     GeneralName,
    roleCertIssuer          [1]     GeneralName,
    roleCertSerialNumber    [2]     CertificateSerialNumber     OPTIONAL,
    roleCertLocator         [3]     GeneralNames                OPTIONAL }

roleSpecCertIdMatch MATCHING-RULE  ::= {
    SYNTAX      RoleSpecCertIdentifierSyntax
    ID          id-mr-roleSpecCertIdMatch }

basicAttConstraints EXTENSION ::=
    {
        SYNTAX              BasicAttConstraintsSyntax
        IDENTIFIED BY       { id-ce-basicAttConstraints }
    }

BasicAttConstraintsSyntax ::= SEQUENCE
    {
        authority           BOOLEAN DEFAULT FALSE,
        pathLenConstraint   INTEGER (0..MAX) OPTIONAL
    }

basicAttConstraintsMatch MATCHING-RULE  ::= {
    SYNTAX      BasicAttConstraintsSyntax
    ID          id-mr-basicAttConstraintsMatch }

delegatedNameConstraints EXTENSION ::= {
    SYNTAX              NameConstraintsSyntax
    IDENTIFIED BY        id-ce-delegatedNameConstraints }

delegatedNameConstraintsMatch MATCHING-RULE  ::= {
    SYNTAX      NameConstraintsSyntax
    ID          id-mr-delegatedNameConstraintsMatch}

timeSpecification EXTENSION  ::=  {
    SYNTAX              TimeSpecification
    IDENTIFIED BY       id-ce-timeSpecification }
```

```
timeSpecificationMatch MATCHING-RULE  ::= {
    SYNTAX      TimeSpecification
    ID          id-mr-timeSpecMatch }


acceptableCertPolicies EXTENSION ::= {
    SYNTAX              AcceptableCertPoliciesSyntax
    IDENTIFIED BY       id-ce-acceptableCertPolicies }

AcceptableCertPoliciesSyntax ::= SEQUENCE SIZE (1..MAX) OF CertPolicyId

CertPolicyId ::= OBJECT IDENTIFIER


acceptableCertPoliciesMatch MATCHING-RULE  ::= {
    SYNTAX      AcceptableCertPoliciesSyntax
    ID          id-mr-acceptableCertPoliciesMatch }

attributeDescriptor EXTENSION  ::= {
    SYNTAX              AttributeDescriptorSyntax
    IDENTIFIED BY       {id-ce-attributeDescriptor } }


AttributeDescriptorSyntax  ::= SEQUENCE {
    identifier                  AttributeIdentifier,
    attributeSyntax             OCTET STRING (SIZE(1..MAX)),
    name                [0]     AttributeName   OPTIONAL,
    description         [1]     AttributeDescription    OPTIONAL,
    dominationRule              PrivilegePolicyIdentifier}


AttributeIdentifier  ::= ATTRIBUTE.&id({AttributeIDs})

AttributeIDs ATTRIBUTE  ::= {...}

AttributeName  ::= UTF8String(SIZE(1..MAX))

AttributeDescription  ::= UTF8String(SIZE(1..MAX))


PrivilegePolicyIdentifier  ::=    SEQUENCE {
    privilegePolicy             PrivilegePolicy,
    privPolSyntax               InfoSyntax }


attDescriptor MATCHING-RULE  ::= {
    SYNTAX      AttributeDescriptorSyntax
    ID          id-mr-attDescriptorMatch }


userNotice  EXTENSION  ::= {
    SYNTAX              SEQUENCE SIZE (1..MAX) OF UserNotice
    IDENTIFIED BY       id-ce-userNotice }


targetingInformation  EXTENSION  ::= {
    SYNTAX              SEQUENCE SIZE (1..MAX) OF Targets
    IDENTIFIED BY       id-ce-targetInformation }


Targets  ::=      SEQUENCE SIZE (1..MAX) OF Target


Target    ::=     CHOICE {
    targetName          [0]         GeneralName,
    targetGroup         [1]         GeneralName,
    targetCert          [2]         TargetCert }


TargetCert        ::=     SEQUENCE {
    targetCertificate   IssuerSerial,
    targetName          GeneralName OPTIONAL,
    certDigestInfo      ObjectDigestInfo OPTIONAL }


noRevAvail EXTENSION ::=  {
    SYNTAX              NULL
    IDENTIFIED BY       id-ce-noRevAvail }
```

```
acceptablePrivilegePolicies EXTENSION ::= {
    SYNTAX              AcceptablePrivilegePoliciesSyntax
    IDENTIFIED BY       id-ce-acceptablePrivilegePolicies }

AcceptablePrivilegePoliciesSyntax       ::=     SEQUENCE SIZE (1..MAX) OF PrivilegePolicy

indirectIssuer EXTENSION ::= {
    SYNTAX      BOOLEAN
    IDENTIFIED BY       id-ce-indirectIssuer }

indirectIssuerMatch MATCHING-RULE  ::= {
    SYNTAX      BOOLEAN
    ID    id-mr-indirectIssuerMatch }

noAssertion EXTENSION ::= {
    SYNTAX NULL
    IDENTIFIED BY id-ce-noAssertion }

issuedOnBehalfOf EXTENSION ::= {
    SYNTAX GeneralName
    IDENTIFIED BY id-ce-issuedOnBehalfOf }

-- 对象标识符赋值 --

-- 对象类别 --

id-oc-pmiUser                       OBJECT IDENTIFIER ::=  {id-oc 24}
id-oc-pmiAA                         OBJECT IDENTIFIER ::=  {id-oc 25}
id-oc-pmiSOA                        OBJECT IDENTIFIER ::=  {id-oc 26}
id-oc-attCertCRLDistributionPts     OBJECT IDENTIFIER ::=  {id-oc 27}
id-oc-privilegePolicy               OBJECT IDENTIFIER ::=  {id-oc 32}
id-oc-pmiDelegationPath             OBJECT IDENTIFIER ::=  {id-oc 33}
id-oc-protectedPrivilegePolicy      OBJECT IDENTIFIER ::=  {id-oc 34}

-- 号码簿属性 --

id-at-attributeCertificate          OBJECT IDENTIFIER ::=  {id-at 58}
id-at-attributeCertificateRevocationList OBJECT IDENTIFIER ::=  {id-at 59}
id-at-aACertificate                 OBJECT IDENTIFIER ::=  {id-at 61}
id-at-attributeDescriptorCertificate OBJECT IDENTIFIER ::=  {id-at 62}
id-at-attributeAuthorityRevocationList OBJECT IDENTIFIER ::=  {id-at 63}
id-at-privPolicy                    OBJECT IDENTIFIER ::=  {id-at 71}
id-at-role                          OBJECT IDENTIFIER ::=  {id-at 72}
id-at-delegationPath                OBJECT IDENTIFIER ::=  {id-at 73}
id-at-protPrivPolicy                OBJECT IDENTIFIER ::=  {id-at 74}
id-at-xMLPrivilegeInfo              OBJECT IDENTIFIER ::=  {id-at 75}
id-at-xMLPprotPrivPolicy            OBJECT IDENTIFIER ::=  {id-at 76}

-- 属性证书扩展 --

id-ce-authorityAttributeIdentifier  OBJECT IDENTIFIER ::=  {id-ce 38}
id-ce-roleSpecCertIdentifier        OBJECT IDENTIFIER ::=  {id-ce 39}
id-ce-basicAttConstraints           OBJECT IDENTIFIER ::=  {id-ce 41}
id-ce-delegatedNameConstraints      OBJECT IDENTIFIER ::=  {id-ce 42}
id-ce-timeSpecification             OBJECT IDENTIFIER ::=  {id-ce 43}
id-ce-attributeDescriptor           OBJECT IDENTIFIER ::=  {id-ce 48}
id-ce-userNotice                    OBJECT IDENTIFIER ::=  {id-ce 49}
id-ce-sOAIdentifier                 OBJECT IDENTIFIER ::=  {id-ce 50}
id-ce-acceptableCertPolicies        OBJECT IDENTIFIER ::=  {id-ce 52}
id-ce-targetInformation             OBJECT IDENTIFIER ::=  {id-ce 55}
id-ce-noRevAvail                    OBJECT IDENTIFIER ::=  {id-ce 56}
id-ce-acceptablePrivilegePolicies   OBJECT IDENTIFIER ::=  {id-ce 57}
id-ce-indirectIssuer                OBJECT IDENTIFIER ::=  {id-ce 61}
id-ce-noAssertion                   OBJECT IDENTIFIER ::=  {id-ce 62}
id-ce-issuedOnBehalfOf              OBJECT IDENTIFIER ::=  {id-ce 64}
```

```
-- PMI匹配规则 --

id-mr-attributeCertificateMatch            OBJECT IDENTIFIER ::= {id-mr 42}
id-mr-attributeCertificateExactMatch       OBJECT IDENTIFIER ::= {id-mr 45}
id-mr-holderIssuerMatch                    OBJECT IDENTIFIER ::= {id-mr 46}
id-mr-authAttIdMatch                       OBJECT IDENTIFIER ::= {id-mr 53}
id-mr-roleSpecCertIdMatch                  OBJECT IDENTIFIER ::= {id-mr 54}
id-mr-basicAttConstraintsMatch             OBJECT IDENTIFIER ::= {id-mr 55}
id-mr-delegatedNameConstraintsMatch        OBJECT IDENTIFIER ::= {id-mr 56}
id-mr-timeSpecMatch                        OBJECT IDENTIFIER ::= {id-mr 57}
id-mr-attDescriptorMatch                   OBJECT IDENTIFIER ::= {id-mr 58}
id-mr-acceptableCertPoliciesMatch          OBJECT IDENTIFIER ::= {id-mr 59}
id-mr-delegationPathMatch                  OBJECT IDENTIFIER ::= {id-mr 61}
id-mr-sOAIdentifierMatch                   OBJECT IDENTIFIER ::= {id-mr 66}
id-mr-indirectIssuerMatch                  OBJECT IDENTIFIER ::= {id-mr 67}


END
```

附 件 B

CRL产生和处理规则

（本附件是本建议书 | 国际标准的组成部分）

B.1 引言

信赖方（证书用户）需要能够对证书的撤消状态进行检查，以便确定是否信任该证书。证书撤消清单（CRL）是一种机制，以便信赖方获得撤消信息。也可以使用其他机制，但在本规范范畴之外。

本附件阐述 CRL 的使用，以便信赖方进行证书撤消状态检查。不同的机构可以有不同的撤消清单发放策略。例如，在某些情况下，证书发放机构可以授权一个不同的机构为其发放的证书发放一个证书撤消清单。某些机构可以将终端实体和 CA 证书的撤消结合进一个单个的清单中，而其他机构可以将这些拆分为单独的清单。某些机构可以将其证书种群分隔为 CRL 片段，某些机构可以在定期的 CRL 间隔之间发布对撤消清单所做的 delta 更新。结果是，出于感兴趣的撤消理由，依据其工作策略，信赖方需要能够确定其所检索的 CRL 的范围，以便它们能够确保拥有完整的、涵盖所议之证书范围的撤消信息集。

编写本附件是为了实现对使用 CRL、完整和彻底终端实体 CRL（EPRLs）以及认证机构撤消清单（CARLs）的公开密钥证书进行撤消状态检查。不过，本描述还可以用于对使用属性证书撤消清单（ACRL）和属性机构撤消清单（AARL）的属性证书进行撤消状态检查。出于本附件的目的，可以认为 ACRL 用于替代 CRL，EPRL 可以是完整和彻底的终端实体 ACRL，AARL 用于替代 CARL。同样，在 B.4 中确定的各号码簿属性将映射于那些有关 AARL 和 ACRL 的属性，以及在发放分发点扩展中用于确定证书类型的各字段可以映射于那些适用于 PMI 的字段。

B.1.1 CRL类型

一个或多个以下类型的 CRL 可供信赖方使用，基于证书发放机构策略的撤消方面：

— 完整的和彻底的 CRL；

— 完整的和彻底的终端实体 CRL（EPRL）；

— 完整的和彻底的认证机构撤消清单（CARL）；

— 分发点 CRL、EPRL 或 CARL；

— 间接的 CRL、EPRL 或 CARL（ICRL）；

— Delta CRL、EPRL 或 CARL；

— 间接的 dCRL、EPRL 或 CARL。

一个完整的和彻底的 CRL 是一个有关所有撤消的终端实体和机构出于某种理由发放的 CA 证书的清单。

一个完整的和彻底的 EPRL 是一个有关机构出于某种理由发放的、所有撤消的终端实体证书的清单。

一个完整的和彻底的 CARL 是一个有关机构出于某种理由发放的、撤消的 CA 证书的清单。

一个分发点 CRL、EPRL 或 CARL 涵盖机构发放的所有证书或证书的一个子集。子集可以依据众多准则。

一个间接的 CRL、EPRL 或 CARL（ICRL）是这样一个 CRL，它包含一个有关撤消的证书的清单，其中的某些证书或全部证书可以不是由签署和发放 CRL 的机构发放。

一个 delta CRL、EPRL 或 CARL 是这样一个 CRL，它只包含对以下 CRL 的更改，即在 dCRL 中参考 CRL 之时对特定的范围而言是完整的 CRL。注意，参考的 CRL 可以是一个对特定的范围而言是完整的 CRL，或者可以是一个用于在本地构建对特定的范围而言是完整的 CRL 的 dCRL。

所有上述 CRL 类型（除了对 dCRL）都是对其特定的范围而言是完整的 CRL 类型。一个 dCRL 将与相关的、对同一范围而言是完整的 CRL 结合使用，以便形成有关证书撤消状态的完整图像。

一个间接的 delta-CRL、EPRL 或 CARL 是这样一个 CRL，它只包含对一个或多个 CRL 集的更改，它们对其特定的范围而言是完整的 CRL，并且其中的某些证书或全部证书可以不是由签署和发放该 CRL 的机构发放。

在本附件以及本规范中，“一个 CRL 的范围”由两个独立的要素来定义。一个要素是 CRL 所涵盖的证书集。另一个要素是 CRL 所涵盖的理由代码集。一个 CRL 的范围可以用一个或多个以下方法来确定：

— 在 CRL 中发放分发点（IDP）扩展；或者

— 在本规范范围之外的其他方法。

B.1.2 CRL处理

如果信赖方使用 CRL 作为确定是否撤消某个证书的机制，那么它们将对该证书使用适当的 CRL。通过遍历诸多特定的步骤，本附件对如何获得和处理适当的 CRL 的过程进行了描述。一个功能上等同于来自本程序之外部行为的执行方案也认为符合本附件和相关规范。某个特殊执行方案用于从特定输入（证书本身以及来自本地策略的输入）获得正确输出（即有关某个证书的撤消状态）的算法不是标准化的。例如，虽然本程序被描述成一系列有待按序处理的步骤，但一个执行方案可以使用在其本地缓冲区中的 CRL，而不是在它每次处理一个证书时对 CRL 进行检索，前提是这些 CRL 对证书范围而言是完整的，不违犯证书或策略的任何参数。

以下一般步骤在下面 B.2-B.5 中描述：

1） 确定有关 CRL 的参数；

2） 确定所需的 CRL；

3） 获得 CRL；

4） 处理 CRL。

步骤 1）从证书和其他地方确定参数，它们将用于确定所需的 CRL 类型；

步骤 2）将参数值用于做出决定；

步骤 3）从可以找到的 CRL 类型来确定号码簿属性；

步骤 4）描述对适当的 CRL 所做的处理。

B.2 确定CRL参数

在证书中内含的信息，以及来自策略的信息，基于这些信息信赖方进行操作，提供了用于判定候选 CRL 正确性的参数。需要以下信息来判定哪个 CRL 类型是适当的：

— 证书类型（即终端实体或 CA）；

— 关键的 CRL 分发点；

— 关键的最新 CRL；

— 兴趣的理由代码。

可以依据证书中的基本约束扩展来确定证书类型。如果扩展出现，那么它指明证书是一个 CA 证书还是一个终端实体证书。如果证书未出现，那么认为证书类型为终端实体证书。本信息用于确定是否可以使用一个 CRL、EPRL 或 CARL 来检查是否撤消证书。

如果证书包含一个关键的 CRL 分发点扩展，那么信赖方证书处理系统将理解本扩展，并将获得和使用由 CRL 分发点扩展为感兴趣的理由代码指向的 CRL，以便确定证书的撤消状态。例如，依靠一个完整的 CRL，将是不够的。

如果证书包含一个关键的、最新的 CRL 扩展，那么若没有首先对最新的 CRL 进行检索和检查，则信赖方不能使用证书。

感兴趣的理由代码由策略来确定，并且通常由应用来提供。建议这些应包括所有的理由代码。本信息用于确定相对理由代码哪些 CRL 是充分的。

注意：当 **freshestCRL** 扩展被标志为非关键时，或者当它未出现在证书中时，策略还可以规定是否希望信赖方对 dCRL 的撤消状态进行检查。虽然排除在本步骤之外，但在步骤 4）中对这些可选 dCRL 的处理进行了描述。

B.3 确定所需的CRL

B.2 中所述的参数值用于确定准则，依据这些准则来确定检查某个特定证书撤消状态所需的 CRL 类型。可以基于以下准则集来确定 CRL 类型，在下面 B.3.1-B.3.4 中描述。

— 带声明的关键 CRL DP 的终端实体证书；
— 带声明的非关键 CRL DP 的终端实体证书；
— 带声明的关键 CRL DP 的 CA 证书；
— 带声明的非关键 CRL DP 的 CA 证书。

在每个子节内对剩余的参数（关键的最新 CRL 扩展以及感兴趣的理由代码集）进行处理。

注意：在每种情况下，有多个 CRL 类型可以满足要求。当存在 CRL 类型可选项时，信赖方可以选择使用任何适当的类型。

B.3.1 带关键CRL DP的终端实体

如果证书是一个终端实体证书，在证书中出现 **cRLDistributionPoints** 扩展，以及标志为关键，那么将获得以下 CRL：

— 来自其中一个已命名分发点 CRL 的 CRL，它涵盖一个或多个感兴趣的理由代码；
— 如果该 CRL 未涵盖所有感兴趣的理由代码，那么剩余理由代码的撤消状态可以通过以下 CRL 的任何组合来满足：
  • 额外的分发点 CRL；
  • 额外的完整的 CRL；
  • 额外的完整的 EPRLs。

如果最新的 CRL 扩展也出现在证书中，并且如果标志为关键的，那么也可以从该扩展的一个或多个已命名分发点中获得一个或多个 CRL，确保该最新的、有关所有感兴趣的理由代码的撤消信息得到检查。

B.3.2 带非关键CRL DP的终端实体

如果证书是一个终端实体证书，在证书中未出现 **cRLDistributionPoints** 扩展，或者出现了 **cRLDistributionPoints** 扩展但未标志为关键，那么可以通过以下 CRL 的任何组合来满足有关所感兴趣的理由代码的撤消状态：

— 分发点 CRL（如果出现的话）；
— 完整的 CRL；
— 完整的 EPRLs。

如果最新的 CRL 扩展也出现在证书中，并且如果标志为关键的，那么也可以从该扩展的一个或多个已命名分发点中获得一个或多个 CRL，确保该最新的、有关所有感兴趣的理由代码的撤消信息得到检查。

B.3.3 带关键CRL DP的CA

如果证书是一个 CA，在证书中出现 **cRLDistributionPoints** 扩展，以及标志为关键，那么将获得以下 CRL/CARLs：

— 来自其中一个已命名分发点的 CRL 或 CARL，它涵盖一个或多个感兴趣的理由代码；
— 如果该 CRL/CARL 未涵盖所有感兴趣的理由代码，那么剩余理由代码的撤消状态可以通过以下 CRL/CARLs 的任何组合来满足：
  • 额外的分发点 CRL/CARLs；
  • 额外的完整的 CRL；
  • 额外的完整的 CARLs。

如果最新的 CRL 扩展也出现在证书中，并且如果标志为关键的，那么也可以从该扩展的一个或多个已命名分发点中获得一个或多个 CRL/CARLs，确保该最新的、有关所有感兴趣的理由代码的撤消信息得到检查。

B.3.4 带非关键CRL DP的CA

如果证书是一个 CA，在证书中未出现 **cRLDistributionPoints** 扩展，或者出现了 **cRLDistributionPoints** 扩展但未标志为关键，那么可以通过以下 CRL 的任何组合来满足有关所感兴趣的理由代码的撤消状态：

— 分发点 CRL/CARLs（如果出现的话）；

— 完整的 CRL；

— 完整的 CARLs。

如果最新的 CRL 扩展也出现在证书中，并且如果标志为关键的，那么也可以从该扩展的一个或多个已命名分发点中获得一个或多个 CRL/CARLs，确保该最新的、有关所有感兴趣的理由代码的撤消信息得到检查。

B.4 获取CRL

如果信赖方正在号码簿中检索适当的 CRL，那么通过检索适当的属性，即一个或多个以下属性，可以从 CRL DP 或证书发放者号码簿条目中获得这些 CRL：

— 证书撤消清单；

— 机构撤消清单；

— Delta 撤消清单。

B.5 处理CRL

在考虑 B.2 中所述的参数、确定 B.3 中所述的适当的 CRL 类型、检索 B.4 中所述的适当的 CRL 集之后，信赖方准备对 CRL 进行处理了。CRL 集将包含至少一个基础 CRL，并可以包含一个或多个 dCRL。对每个要处理的 CRL，信赖方将确保 CRL 的范围是合适的。经过上面 B.2 和 B.3 中所述的处理，信赖方已经确定 CRL 对感兴趣的证书范围而言是合适的。另外，应对 CRL 进行有效性检查，通过检查来确定证书是否已被撤消。这些检查在下面 B.5.1-B.5.4 中描述。

B.5.1 验证基础CRL范围

如 B.3 中所述，可以有多种 CRL 类型可以用作基础 CRL，用于检查证书的撤消状态。依据发放机构有关 CRL 发放的策略，对信赖方可以有一个或多个以下基础 CRL 类型可用。

— 对所有实体而言完整的 CRL；

— 完整的 EPRL；

— 完整的 CARL；

— 基于分发点的 CRL/EPRL/CARL。

子节 B.5.1.1-B.5.1.4 提供了将为 TRUE 的条件集，以便信赖方使用一个每种类型的 CRL 作为基础 CRL，出于感兴趣的理由代码，用于检查证书撤消状态。

间接的基础 CRL 在每个子节中进行描述。

B.5.1.1 完整的CRL

为了确定 CRL 对 CRL 发放者所负责的终端实体和 CA 证书是一个完整的 CRL，对所有感兴趣的理由代码，以下都将为 TRUE：

— Delta CRL 指示符扩展将不出现；以及

— 发放分发点扩展可以出现；以及

— 要么发放分发点扩展不得包含分发点字段，要么分发点字段中的其中一个名称应匹配于 CRL 中的 **issuer** 字段；以及

— 发放分发点扩展要么不包含任何以下字段，要么如果它包含任何以下字段，那么任何出现的字段都不得设为 TRUE：containsUserPublicKeyCerts、containsCACerts、containsUserAttributeCerts、containsAACerts 与/或 containsSOAPublicKeyCerts；以及

— 如果 **reasonCodes** 字段出现在发放分发点扩展中，那么理由代码字段将包括对应用感兴趣的所有理由；以及

— 发放分发点扩展可以包含也可以不包含 **indirectCRL** 字段（因此，无需对该字段进行检查）。

B.5.1.2 完整的EPRL

为了确定 CRL 对感兴趣的理由代码是一个完整的 EPRL，所有以下都将为 TRUE：

— Delta CRL 指示符扩展将不出现；以及

— 发放分发点扩展将出现；以及

— 要么发放分发点扩展不得包含分发点字段，要么分发点字段中的其中一个名称应匹配于 CRL 中的 **issuer** 字段；以及

— 发放分发点扩展将包含 **containsUserPublicKeyCerts** 字段，该字段将设为 **TRUE**；以及

— 如果 **reasonCodes** 字段出现在发放分发点扩展中，那么理由代码字段将包括对应用感兴趣的所有理由；以及

— 发放分发点扩展可以包含也可以不包含 **indirectCRL** 字段（因此，无需对该字段进行检查）。

本 CRL 只有当信赖方已确定对象证书是一个终端实体证书时才能使用。因此，如果对象证书包含 **basicConstraints** 扩展，那么其值将为 **cA=FALSE**。

B.5.1.3 完整的CARL

为了确定 CRL 对感兴趣的理由代码是一个完整的 CARL，所有以下条件都将为 TRUE：

— Delta CRL 指示符扩展将不出现；以及

— 发放分发点扩展将出现；以及

— 要么发放分发点扩展不得包含分发点字段，要么分发点字段中的其中一个名称应匹配于 CRL 中的 **issuer** 字段；以及

— 发放分发点将包含 **containsCACerts** 字段，该字段将设为 **TRUE**；以及

— 如果 **reasonCodes** 字段出现在发放分发点扩展中，那么理由代码字段将包括对应用感兴趣的所有理由；以及

— 发放分发点扩展可以包含也可以不包含 **indirectCRL** 字段（因此，无需对该字段进行检查）。

本 CARL 只有当对象证书是一个 CA 证书时才能使用。因此，对象证书将包含值为 **cA=TRUE** 的 **basicConstraints** 扩展。

B.5.1.4 基于CRL/EPRL/CARL的分发点

为了确定 CRL 是 CRL 分发点扩展或证书中最新的 CRL 扩展所指明的 CRL 之一，所有以下条件都将为 TRUE：

— 要么 CRL 发放分发点扩展中的分发点字段不出现（只有当不寻找关键的 CRL DP 时），要么 CRL 分发点扩展或证书最新 CRL 扩展的分发点字段中的其中一个名称应匹配于 CRL 发放分发点扩展的分发点字段中的其中一个名称。可选地，证书 CRL DP 或最新 CRL 扩展的 **cRLIssuer** 字段中的其中一个名称可以匹配于 IDP 的 DP 中的其中一个名称；以及

— 发放分发点扩展要么不包含任何以下字段，要么如果它包含任何以下字段，那么任何出现的字段都不得设为 TRUE：containsUserPublicKeyCerts、containsCACerts、containsUserAttributeCerts、containsAACerts 与/或 containsSOAPublicKeyCerts，要么适合证书类型的字段设为 TRUE（请参见表 B.1，关于每种证书类型的字段类型）；以及

— 如果理由代码字段出现在 CRL 分发点扩展或证书的最新 CRL 扩展中，那么该字段要么不出现在 CRL 的发放分发点扩展中，要么包含至少一个在证书的 CRL 分发点扩展中声明的理由代码；以及

— 如果 **cRLIssuer** 字段未出现在证书的 CRL 分发点扩展中，那么将由签署证书的同一 CA 来签署 CRL；以及

— 如果 **cRLIssuer** 字段出现在证书的相对扩展（CRL 分发点或最新的 CRL 扩展）中，那么 CRL 将由 CRL 分发点扩展或证书最新 CRL 扩展中确定的 CRL 发放者来签署，CRL 将在发放分发点扩展中包含 **indirectCRL** 字段。

注 — 当对理由和字段是否出现进行测试时，只有当字段出现在 CRL DP 的同一 DistributionPoint 或最新的 CRL 扩展中时，测试才会成功，对这，在相应 CRL 中的 IDP 扩展的分发点字段中，存在一个名称匹配。

表 B.1— 证书类型和发放分发点字段

证书类型	发放分发点字段
终端实体（公开密钥）	containsUserPublicKeyCerts
CA	containsCACerts
终端实体（属性）	containsUserAttributeCerts
AA	containsAACerts
SOA	containsSOAPublicKeyCerts

B.5.2 验证delta CRL范围

信赖方还可以对 dCRL 进行检查，因为需要通过一个证书或 CRL 中的关键 **freshestCRL** 扩展，或者因为信赖方操作所依据的策略要求进行 dCRL 检查。

如果所有以下条件都满足，那么信赖方总是可以相信，它将拥有适当的、有关证书的 CRL 信息：

— 信赖方使用的基础 CRL 对证书而言是适当的（依据范围）；以及

— 信赖方使用的 delta CRL 对证书而言是适当的（依据范围）；以及

— 在 dCRL 参考基础 CRL 之时或之后发放基础 CRL。

为了确定 dCRL 对证书合适，所有以下条件都将为 TRUE：

— Delta CRL 指示符扩展将出现；以及

— dCRL 将在基础 CRL 之后发放。确保它的一种方法是检查 dCRL **crlNumber** 扩展中的 CRL 数量大于信赖方所用基础 CRL **crlNumber** 扩展以及基础和 dCRL 匹配中 **cRLStreamIdentifier** 字段中的 CRL 数量。该方法可能需要额外的逻辑来解决数量限制问题。另一种方法是对基础 CRL 的 **thisUpdate** 字段和信赖方拥有的 dCRL 的 **thisUpdate** 字段进行比较；以及

— 确保它的一种方法是检查 dCRL **deltaCRLIndicator** 扩展中的 CRL 数量小于或等于信赖方所用基础 CRL **crlNumber** 扩展以及基础和 dCRL 匹配中 **cRLStreamIdentifier** 字段中的 CRL 数量。该方法可能需要额外的逻辑来解决数量限制问题。另一种方法是对信赖方拥有的基础 CRL 的 **thisUpdate** 字段和 dCRL 指向的基础 CRL 的 t**hisUpdate** 字段进行比较。还有一种方法是对信赖方拥有的基础 CRL 的 **thisUpdate** 字段和信赖方拥有的 dCRL 的 **baseUpdateTime** 扩展进行比较；以及

注 — 只要利用 **crlNumber** 和 **cRLStreamIdentifier** 检查，使上述两个规则得到满足，那么一个信赖方就总能通过将 dCRL 应用于基础 CRL 来构建一个基础 CRL。在这种情况下，新的、基础 CRL 的 **crlNumber** 扩展和 **thisUpdate** 字段就为 dCRL 的扩展和字段。信赖方不知道新的、基础 CRL 的 **nextUpdate** 字段，并且无需知道将其与另一个 dCRL 相关联的目的。

— 如果 dCRL 包含一个发放分发点扩展，那么发放分发点的范围将与上面 B.5.1.4 中所述的证书相一致；以及

— 如果 dCRL 不包含任何以下扩展：**streamIdentifier** 和 **issuingDistributionPoint**，那么它将只能与一个完整的、彻底的基础 CRL 结合使用。

B.5.3 基础CRL的有效性和流通检查

为了验证基础 CRL 是准确的，并且自其发放后未经修改，所有以下条件都应满足：

— 利用经过认证的方法，信赖方将可获得在 CRL 中确定的发放者公开密钥；以及

— 将利用该经过认证的公开密钥对基础 CRL 上的签名进行验证；以及

— 如果 **nextUpdate** 字段出现，那么当前时间应在 **nextUpdate** 字段之前；以及

— CRL 中的发放者名称应匹配于证书中、正在检查是否撤消的发放者名称，除非 CRL 检索自证书中的 CRL DP，并且 CRL DP 扩展包含 CRL 发放者部件。在这种情况下，CRL DP 扩展的 CRL 发放者部件中的其中一个名称应匹配于 CRL 中的发放者名称。

B.5.4 delta CRL的有效性和检查

为了验证 dCRL 是准确的，并且自其发放后未经修改，所有以下条件都应满足：

— 利用经过认证的方法，信赖方将可获得在 CRL 中确定的发放者公开密钥；以及

— 将利用该经过认证的公开密钥对 dCRL 上的签名进行验证；以及

— 如果 **nextUpdate** 字段出现，那么当前时间应在 **nextUpdate** 字段之前；以及

— dCRL 中的发放者名称应匹配于证书中、正在检查是否撤消的发放者名称，除非 Delta CRL 检索自证书中的 CRL DP，并且 CRL DP 扩展包含 CRL 发放者部件。在这种情况下，CRL DP 扩展的 CRL 发放者部件中的其中一个名称应匹配于 CRL 中的发放者名称。

附 件 C

delta CRL发布举例

（本附件不是本建议书 | 国际标准的组成部分）

有两种发放 CRL 模型，涉及对特定的证书集使用 dCRL。

在第一个模型中，每个 dCRL 都参考最新的 CRL，它对特定的范围是完整的。在发放对该范围而言是完整的、新的 CRL 之前，对同一范围可以发放若干个 dCRL。对该范围而言是完整的、新的 CRL 用作为下一个 dCRL 序列的基础，并且是在 dCRL 的相关扩展中参考的那个 CRL。当发放对该范围而言是完整的、新的 CRL 时，还将发放对该范围而言是完整的、前一个 CRL 的最后一个 dCRL。

第二个模型非常相似，不同的是 dCRL 参考的 CRL 对特定的范围而言不必是完整的（即参考的 CRL 可以只作为一个 dCRL 发放）。如果参考的 CRL 对特定的范围而言是完整的，那么它可以不必是最新的、对特定的范围而言是完整的 CRL。

使用系统处理 dCRL 的证书还需拥有一个对特定的范围而言是完整的 CRL，并且其新旧程度应至少同 dCRL 中参考的 CRL。对特定的范围而言是完整的这个 CRL 可以由负责的机构发放，或者可以由使用证书的系统在本地构建。注意，在某些情况下，在对特定的范围而言是完整的 dCRL 和 CRL 中，可以有完全相同的信息，例如，当使用证书的系统有一个在 dCRL 中所参考的 CRL 之后发放的 CRL 时。

下表描述了有关 dCRL 使用的三个例子。例 1 是传统的方案，即上面所述的第一个模型。例 2 和例 3 是上面所述的第二个模型的变种。

在例 2 中，机构发放 CRL，它们对特定的范围、每个第二天而言都是完整的，dCRL 参考第二天到最后一天，对范围 CRL 而言是完整的。该方案在以下环境中是有用的，即需要压缩同时访问知识库的用户数量，以检索对特定的范围而言是完整的 CRL。在例 2 中，拥有最新的、对该范围而言是完整的 CRL 的用户，以及拥有第二新的、对该范围而言是完整的 CRL 的用户，可以使用相同的 dCRL。两个用户集都拥有完整的撤消信息，这些信息有关所用 dCRL 发放之时、对该特定范围的证书。

在例 3 中，像例 1 中一样，对特定的范围而言是完整的 CRL 每星期发放一次，但每个 dCRL 参考一个比该 dCRL 早 7 天的撤消信息库。

在此未提供有关间接 CRL 使用的例子，但它是一个有关这些例子的超集。

这些只是例子，其他变种也是可能的，这依赖于本地策略。在建立该策略时可能需要考虑到的一些因素包括：用户数量、访问 CRL 的频度、CRL 的复制、持有 CRL 的号码簿系统的负载平衡、性能、等待时间要求等。

时间	例1 – Delta参考最新的、对特定范围完整的CRL		例2 – Delta参考第二新的、对特定范围完整的CRL		例3 – Delta参考已过7天的撤消信息	
	对特定范围完整的CRL	Delta-CRL	对特定范围完整的CRL	Delta-CRL	对特定范围完整的CRL	Delta-CRL
8	thisUpdate = 第 8 天 nextUpdate = 第 15 天 crlNumber = 8	thisUpdate = 第 8 天 nextUpdate = 第 9 天 crlNumber = 8 BaseCRLNumber = 1	thisUpdate = 第 8 天 nextUpdate = 第 10 天 crlNumber = 8	thisUpdate = 第 8 天 nextUpdate = 第 9 天 crlNumber = 8 BaseCRLNumber = 6	thisUpdate = 第 8 天 nextUpdate = 第 15 天 cRLNumber = 8	thisUpdate = 第 8 天 nextUpdate = 第 9 天 cRLNumber = 8 BaseCRLNumber = 1
9	未发布	thisUpdate = 第 9 天 nextUpdate = 第 10 天 crlNumber = 9 BaseCRLNumber = 8	未发布	thisUpdate = 第 9 天 nextUpdate = 第 10 天 crlNumber = 9 BaseCRLNumber = 6	未发布	thisUpdate = 第 9 天 nextUpdate = 第 10 天 cRLNumber = 9 BaseCRLNumber = 2
10	未发布	thisUpdate = 第 10 天 nextUpdate = 第 11 天 crlNumber = 10 BaseCRLNumber = 8	thisUpdate = 第 10 天 nextUpdate = 第 12 天 crlNumber = 10	thisUpdate = 第 10 天 nextUpdate = 第 11 天 crlNumber = 10 BaseCRLNumber = 8	未发布	thisUpdate = 第 10 天 nextUpdate = 第 11 天 cRLNumber = 10 BaseCRLNumber = 3
11-14	延续有关之前日子的样式					
15	thisUpdate = 第 15 天 nextUpdate = 第 22 天 crlNumber = 15	thisUpdate = 第 15 天 nextUpdate = 第 16 天 crlNumber = 15 BaseCRLNumber = 8	未发布	thisUpdate = 第 15 天 nextUpdate = 第 16 天 crlNumber = 15 BaseCRLNumber = 12	thisUpdate = 第 15 天 nextUpdate = 第 22 天 cRLNumber = 15	thisUpdate = 第 15 天 nextUpdate = 第 16 天 cRLNumber = 15 BaseCRLNumber = 8
16	未发布	thisUpdate = 第 16 天 nextUpdate = 第 17 天 crlNumber = 16 BaseCRLNumber = 15	thisUpdate = 第 16 天 nextUpdate = 第 18 天 crlNumber = 16	thisUpdate = 第 16 天 nextUpdate = 第 17 天 crlNumber = 16 BaseCRLNumber = 14	未发布	thisUpdate = 第 16 天 nextUpdate = 第 17 天 cRLNumber = 16 BaseCRLNumber = 9

附 件 D

特权策略和特权属性定义举例

（本附件不是本建议书 | 国际标准的组成部分）

D.1 概述

特权策略为特权管理准确地定义了何时特权验证者应对提交的特权集是否足够做出结论，以便它授权特权声明者可以访问（请求的对象、资源、应用等）。特权策略的正式规范可以帮助特权验证者依据所请求资源的敏感性自动评估特权声明者的特权，原因是，它包括依据其特权和资源敏感性来判定是否同意特权声明者请求的规则。

由于需要确保在这些判定中所用特权策略的完整性，因此在经签署的对象中可以传达一个以对象标识符形式存在的特权策略标识符，以及一个有关整个特权策略的 HASH，并保存在号码簿条目中，等等。不过，在本规范中没有对任何用于定义特权策略实例的特定语法进行定义。

D.2 样本句法

特权策略可以利用任何句法来定义，包括纯文本。为了帮助那些定义特权策略的人员能够更好地理解各种不同的定义选项，出于此目的，在本附件中提供了两个语法例子。需要强调的是：这些只是例子，通过使用属性证书或公开密钥证书的 **subjectDirectoryAttributes** 扩展来执行特权管理，并不要求支持这些语法或任何其他特定的语法。

D.2.1 第一个例子

以下 ASN.1 句法是有关综合、灵活的特权策略定义工具的一个例子。

```
PrivilegePolicySyntax     ::= SEQUENCE {
    version       Version,
    ppe           PrivPolicyExpression }

PrivPolicyExpression      ::= CHOICE {
    ppPredicate [0] PrivPolicyPredicate,
    and           [1] SET SIZE (2..MAX) OF PrivPolicyExpression,
    or            [2] SET SIZE (2..MAX) OF PrivPolicyExpression,
    not           [3] PrivPolicyExpression,
    orderedPPE [4] SEQUENCE OF PrivPolicyExpression }
    -- 注：“序列”定义了对特权进行检查的
    -- 临时顺序。

PrivPolicyPredicate       ::= CHOICE {
    present               [0] PrivilegeIdentifier,
    equality              [1] PrivilegeComparison, -- 单一/集合值的特权。
    greaterOrEqual        [2] PrivilegeComparison, -- 单一值的特权。
    lessOrEqual           [3] PrivilegeComparison, -- 单一值的特权。
    subordinate           [4] PrivilegeComparison, -- 单一值的特权。
    substrings            [5] SEQUENCE {           -- 单一值的特权。
        type                      PrivilegeType,
        initial               [0] PrivilegeValue OPTIONAL,
        any                   [1] SEQUENCE OF PrivilegeValue,
        final                 [2] PrivilegeValue OPTIONAL },
subsetOf                  [6] PrivilegeComparison, -- 集合值的特权。
supersetOf                [7] PrivilegeComparison, -- 集合值的特权。
nonNullSetInter           [8] PrivilegeComparison, -- 集合值的特权。
approxMatch               [9] PrivilegeComparison,
-- 单一/集合值的特权。（近似值由应用定义）
extensibleMatch           [10] SEQUENCE {
        matchingRule          OBJECT IDENTIFIER,
        inputs                PrivilegeComparison } }


PrivilegeComparison       ::= CHOICE {
    explicit                  [0] Privilege,
```

```
    -- 由Privilege.privilegeId 确定的外部特权的值与
    -- 在Privilege.privilegeValueSet 中显性提供的值
    -- 相比较。
    byReference          [1] PrivilegeIdPair }
    -- 由PrivilegeIdPair.firstPrivilege 确定的外部特权的值与
    -- 由PrivilegeIdPair.secondPrivilege确定的第二个外部特权的值
    -- 相比较。

Privilege        ::= SEQUENCE {
    type         PRIVILEGE.&id ({SupportedPrivileges}),
    values       SET SIZE (0..MAX) OF
                 PRIVILEGE.&Type ({SupportedPrivileges} {@type})
    }


SupportedPrivileges  PRIVILEGE ::=      { ... }
PRIVILEGE        ::=      ATTRIBUTE
-- 特权类似属性。

PrivilegeIdPair  ::=          SEQUENCE {
    firstPrivilege            PrivilegeIdentifier,
    secondPrivilege           PrivilegeIdentifier }

PrivilegeIdentifier     ::=   CHOICE {
    privilegeType             [0] PRIVILEGE.&id ({SupportedPrivileges}),
    xmlTag                    [1] OCTET STRING,
    edifactField              [2] OCTET STRING }
-- PrivilegeIdentifier将AttributeType的概念延伸至了
-- 其他（如经过标记的）环境中，如XML和EDIFACT。

Version                  ::= INTEGER { v1(0) }
```

一个具体的例子可能有助于说明上述 **PrivilegePolicy** 构件如何创建和使用。

考虑批准提高薪金的特权。为简化起见，假设执行状态的策略，即只有高级经理和上述人员可以批准提高薪金，以及批准只能提供给低于你自身职位的人员（例如，一个董事可以批准为高级经理提高薪金，但不可以批准为副总裁提高薪金）。对这个例子，假设有六种可能的职级（“技术人员”＝0，“经理”＝1，“高级经理”＝2，“董事”＝3，“副总裁”＝4，“总裁”＝5）。

进一步假设，属性证书中确定职级的属性类型（“特权”）为 OBJECT ID *OID-C*，数据库记录中确定职级的属性类型（“灵敏度”）其薪金字段将改为 OBJECT ID *OID-D*（当然，这些将由实际执行中的真实对象标识符所替代）。以下布尔表达式表示期望的“薪金批准”策略（在 **PrivilegePolicy** 表达式中将其编为代码是一个相对直接的任务）：

AND (NOT (lessOrEqual (value corresponding to OID-C, value corresponding to OID-D))

subsetOf (value corresponding to OID-C, { 2, 3, 4, 5 }))

策略编码指出，批准者的职级应大于（表示为“NOT less-than-or-equal-to”）被批准者的职级，以及批准者的职级应为{高级经理，……，总裁}中之一，以便本布尔表达式的值为 TRUE。第一个特权比较为“通过参考”，对所涉的两个实体比较对应属性类型“职级”的值。第二个特权比较为“显性地”；在此，对批准者对应特权“职级”的值与显性包含的值清单进行比较。因此，在这种情况下，特权验证者需要一个构件来对本策略以及两个属性进行编码，一个属性与批准者相关，另一个属性与被批准者相关。批准者的属性（它将包含在属性证书中）值可以是{*OID-C* 3}，被批准者的属性（它可以包含在数据库记录中）值可以是{*OID-D* 3}。对对应批准者属性类型（在本例中为 3）的属性值与对应被批准者属性类型（在本例中也为 3）的属性值进行比较，结果是，对“NOT lessOrEqual”表达式的比较结果为 FALSE，因此第一个董事无权批准为第二个董事提高薪金。另一方面，如果被批准者的属性为{*OID-D* 1}，那么董事将有权批准为经理提高薪金。

不难设想可以为以上表达式增加有用的额外内容。例如，可以增加第三个部件“**and**”，用以表示环境变量“currentTime”，从本地时钟读取，而后编码为一个 OBJECT ID *OID-E* 类型的属性，将落在表达式明确规定的某个特殊范围内，作为一个 OBJECT ID *OID-F* 类型的属性。因此，例如，只有满足上述条件并且请求发生在工作时间内时，才允许更新薪金。

D.2.2 第二个例子

安全策略的最简单形式是一个准则集，用于提供安全服务。关于访问控制，安全策略是更高的系统级安全策略的一个子集，它定义了在发起者和目标之间执行访问控制策略的方法。访问控制机制需要允许特定策略许可的通信；拒绝特定策略未明确允许的通信。

安全策略是访问控制机制进行决策的基础。域特定的安全策略信息通过安全策略信息文件（SPIF）来传达。

批注 [P146]: Page: 133
A: Security Policy Information File (SPIF)

SPIF 是一个经签署的对象，使之免受未经授权的更改。SPIF 包含用于解释安全标签和许可证属性中所含访问控制参数的信息。出现在许可证属性中的安全策略标识符需要与安全策略中所定义的某个特定执行语法和语义相关。与特定安全策略相关的该执行语法在 SPIF 中维护。

SPIF 通过由安全策略确定的安全策略域来在机构和敏感度之间传达等同性，它为安全标签提供了一种可印刷的表示方式，并将可显示的字符串映射至安全级别和类别，以便在选择数据对象的安全属性时能呈现给最终用户。等同性映射以这样一种方式表示，是为了使在一个安全策略域下产生的标签可以被工作于另一个安全策略域中的应用所正确理解。SPIF 还将许可证属性映射于消息安全标签字段，表示标签应显示给用户。如果成功的话，这种映射将确认计划中的接收者有适当的授权来接受数据对象。

一个 SPIF 包含一系列以下内容：

- **versionInformation** — 指明 ASN.1 句法的版本。
- **updateInformation** — 指明句法的版本以及 SPIF 规范的语义。
- **securityPolicyIdData** — 确定 SPIF 所用的安全策略。
- **privilegeId** — 指明确定句法的 OID，它包括在许可证属性安全类别中。
- **rbacId** — 对象标识符，它与 SPIF 一起用于确定安全类别的句法。
- **securityClassifications** — 将安全标签分类映射于许可证属性中的分类，并且还提供等同映射。
- **securityCategoryTagSets** — 将安全标签类别映射于许可证属性中的安全类别，并且还提供等同映射。
- **equivalentPolicies** — 巩固 SPIF 中的所有等同策略。
- **defaultSecurityPolicyIdData** — 如果接受的数据不带安全标签，那么确定将要使用的安全策略。
- **extensions** — 提供一种机制，以便纳入用于确定未来要求的额外功能。

安全策略信息文件用下列句法进行定义：

```
SecurityPolicyInformationFile ::= SIGNED {SPIF}
SPIF ::=  SEQUENCE    {
    versionInformation              VersionInformationData DEFAULT v1,
    updateInformation               UpdateInformationData,
    securityPolicyIdData                    ObjectIdData,
    privilegeId                             OBJECT IDENTIFIER,
    rbacId                                  OBJECT IDENTIFIER,
    securityClassifications         [0]     SEQUENCE OF SecurityClassification OPTIONAL,
    securityCategories              [1]     SEQUENCE OF SecurityCategory  OPTIONAL,
    equivalentPolicies              [2]     SEQUENCE OF EquivalentPolicy  OPTIONAL,
    defaultSecurityPolicyIdData     [3]     ObjectIdData OPTIONAL,
    extensions                      [4]     Extensions OPTIONAL }

VersionInformationData ::= INTEGER { v1(0) }


UpdateInformationData  ::=  SEQUENCE    {
    sPIFVersionNumber               INTEGER,
    creationDate                    GeneralizedTime,
    originatorDistinguishedName     Name,
    keyIdentifier                   OCTET STRING OPTIONAL }

ObjectIdData   ::=   SEQUENCE  {
    objectId            OBJECT IDENTIFIER,
    objectIdName        DirectoryString {ubObjectIdNameLength} }
```

```
SecurityClassification ::=  SEQUENCE  {
    labelAndCertValue                   INTEGER,
    classificationName                  DirectoryString {ubClassificationNameLength},
    equivalentClassifications   [0]     SEQUENCE OF EquivalentClassification OPTIONAL,
    hierarchyValue                      INTEGER,
    markingData                 [1]     SEQUENCE OF MarkingData OPTIONAL,
    requiredCategory            [2]     SEQUENCE OF OptionalCategoryGroup OPTIONAL,
    obsolete                            BOOLEAN DEFAULT FALSE           }

EquivalentClassification ::= SEQUENCE {
    securityPolicyId            OBJECT IDENTIFIER,
    labelAndCertValue           INTEGER,
    applied                     INTEGER {
                                    encrypt     (0),
                                    decrypt     (1),
                                    both        (2) } }

MarkingData ::= SEQUENCE {
    markingPhrase       DirectoryString {ubMarkingPhraseLength} OPTIONAL,
    markingCodes        SEQUENCE OF MarkingCode OPTIONAL }

MarkingCode ::= INTEGER {
    pageTop                 (1),
    pageBottom              (2),
    pageTopBottom           (3),
    documentEnd             (4),
    noNameDisplay           (5),
    noMarkingDisplay        (6),
    unused                  (7),
    documentStart           (8),
    suppressClassName       (9)}

OptionalCategoryGroup ::= SEQUENCE {
    operation               INTEGER {
                                onlyOne         (1),
                                oneOrMore       (2),
                                all             (3)},
    categoryGroup       SEQUENCE OF OptionalCategoryData }

OptionalCategoryData ::= SEQUENCE {
    optCatDataId            OC-DATA.&id({CatData}),
    categorydata            OC-DATA.&Type({CatData}{@optCatDataId }) }

OC-DATA ::= TYPE-IDENTIFIER

CatData OC-DATA ::= {  ... }

EquivalentPolicy        ::= SEQUENCE {
    securityPolicyId        OBJECT IDENTIFIER,
    securityPolicyName      DirectoryString {ubObjectIDNameLength}
    OPTIONAL}

Extensions ::= SEQUENCE OF Extension

Extension ::= SEQUENCE {
    extensionId             EXTENSION.&objId ({ExtensionSet}),
    critical                BOOLEAN DEFAULT FALSE,
    extensionValue              OCTET STRING  }
```

注意：SPIF 例子是一种不断进化演变的语法，有关各个元素的完整定义和描述可以在 ITU-T X.841 建议书 | ISO/IEC 15816 中找到。

D.3 特权属性举例

用于表达某个特定特权的属性的以下例子纯粹只是为了描述之用。本语法的实际规范以及相关的属性包含在 ITU-T X.501 建议书 | ISO/IEC 9594-2 的第 19.5 节中。本特殊属性用于表达与某个已命名的实体相关的许可证，包括用于与 DSA 进行通信目的的 DUA。

许可证属性将许可证与一个包括 DUA 的、已命名的实体相关联。

```
clearance  ATTRIBUTE ::=  {
     WITH SYNTAX              Clearance
     ID                       id-at-clearance }

Clearance  ::=  SEQUENCE {
     policyId                 OBJECT IDENTIFIER,
     classList                ClassList DEFAULT {unclassified},
     securityCategories       SET SIZE (1MAX) OF SecurityCategory  OPTIONAL}

ClassList  ::=  BIT STRING {
     unmarked            (0),
     unclassified        (1),
     restricted          (2),
     confidential        (3),
     secret              (4),
     topSecret           (5) }
```

在参考文献中利用本特权的实际规范对各部件进行了描述。

附 件 E

公开密钥密码系统概述[3)]

（本附件不是本建议书 | 国际标准的组成部分）

在传统的加密系统中，秘密消息发起者用于加密信息的密钥与合法的接收者用于解密消息的密钥是相同的。

不过，在公开密钥加密系统（PKCS）中，密钥是成对出现的，其中的一个密钥用于加密，另一个密钥用于解密。每个密钥对与某个特定的用户 X 相关。其中一个密钥，即公开密钥（Xp），众所周知，任何用户都可以用它来对数据进行加密。只有拥有互补的专用密钥（Xs）的 X 才能解密数据。（记法上，这用 D = Xs[Xp[D]]来表示。）计算上，无法从专用密钥获得有关公开密钥的知识。因此，通过 Xp 进行加密，任何用户都可以传递只有 X 才能解开的信息片段。通过扩展，通过相互利用彼此的公开密钥来加密数据，两个用户就可以进行秘密通信了，如图 E.1 所示。

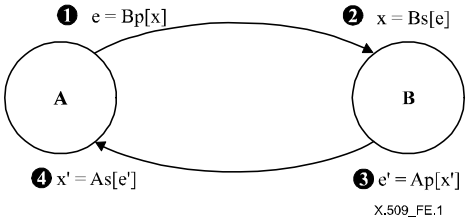


图 E.1— 使用PKCS来交换秘密信息

用户 A 拥有公开密钥 Ap 和专用密钥 As，用户 B 拥有另一个密钥集 Bp 和 Bs。A 和 B 相互都知道彼此的公开密钥，但都不知道另一方的专用密钥。A 和 B 因此可以通过以下步骤（如图 E.1 所示），来相互交换秘密信息：

1） A 希望将某些秘密信息 x 传送给 B。因此，A 利用 B 的加密密钥对 x 进行加密，并将加密后的信息 e 传送给 B。这用以下公式表示：

$$e = Bp[x]$$

2） B 现在可以利用秘密解密密钥 Bs 对密码 e 进行解密，以获得信息 x。注意：B 是 Bs 的惟一拥有者，由于该密钥从不允许透露或传送，因此任何其他方都不可能获得信息 x。拥有 Bs 确定了 B 的身份。解密操作用以下公式表示：

$$x = Bs[e]或x = Bs[Bp[x]]$$

3） 同样，B 现在可以利用 A 的加密密钥 Ap 加密后将某些秘密信息 x'传送给 A：

$$e' = Ap[x']$$

4） A 通过对 e'进行解密，以获得 x'：

$$x' = As[e']或x' = As[Ap[x']]$$

---

3) 更多的信息，参见：

DIFFIE（W.）和HELLMAN（M.E.）：New Directions in Cryptography，*IEEE Transactions on Information Theory*，IT-22，No. 6，1976年11月。

通过这种方法，A 和 B 实现了秘密信息 x 和 x'的交换。除了 A 和 B 之外，该信息不能被任何其他方获得，前提是其秘密密钥未被透露。

这样一种交换以及在各方之间传送秘密信息，可以用于验证其身份。尤其是，通过分别验证是否拥有秘密加密密钥 As 和 Bs，可以来验证 A 和 B 的身份。通过在 B 的消息 x'中返回部分其信息 x，A 就可以判定 B 是否拥有秘密加密密钥 Bs。这向 A 表明，与 Bs 的拥有者进行了通信。B 也可以利用同样的方法来对 A 的身份进行测试。

解密和加密步骤可以颠倒是某些 PKCS 的一个特性，如在 D = Xp[Xs[D]]中。这允许任何用户（拥有 Xp）读取只能由 X 产生的信息片段。因此，可以用之来认证信息源，这是数字签名的基础。只有拥有该（转置）特性的 PKCS 才适合在本鉴权框架中使用。在附件 D 中对这样一个算法进行了描述。

附 件 F

算法对象标识符的参考定义

（本附件是本建议书 | 国际标准的组成部分）

本附件定义了在没有正式寄存器的情况下指派给鉴权和加密算法的对象标识符。它旨在当其变得可用时使用这样一个寄存器。各定义采用了 ASN.1 模块形式，“**AlgorithmObjectIdentifiers**”。

```
AlgorithmObjectIdentifiers {joint-iso-itu-t ds(5) module(1) algorithmObjectIdentifiers(8) 5}
DEFINITIONS ::=
BEGIN

-- EXPORTS All --
-- 输出在该模块中规定的类型和值，用于本号码簿规范中涵盖的其他ASN.1模块，还要使用它们接入到号码簿
-- 业务的其他应用中。其他的应用可以把它们用于自己的目的，但这并不会限制为维护或改进号码簿业务所需
-- 的扩展和修改。

IMPORTS
    algorithm, authenticationFramework
        FROM UsefulDefinitions {joint-iso-itu-t ds(5) module(1) usefulDefinitions(0) 5}
    ALGORITHM
        FROM AuthenticationFramework authenticationFramework;


-- 对象标识符类别 --

encryptionAlgorithm     OBJECT IDENTIFIER     ::=     {algorithm 1}
hashAlgorithm           OBJECT IDENTIFIER     ::=     {algorithm 2}
signatureAlgorithm      OBJECT IDENTIFIER     ::=     {algorithm 3}


-- 同义词 --

id-ea     OBJECT IDENTIFIER     ::=     encryptionAlgorithm
id-ha     OBJECT IDENTIFIER     ::=     hashAlgorithm
id-sa     OBJECT IDENTIFIER     ::=     signatureAlgorithm


-- 算法 --

rsaALGORITHM ::=      {
    KeySize
    IDENTIFIED BY     id-ea-rsa }

KeySize          ::=      INTEGER


-- 以下对象标识符赋值用于保存指派给不赞成功能的值。

id-ea-rsa                  OBJECT IDENTIFIER     ::=     {id-ea 1}
id-ha-sqMod-n              OBJECT IDENTIFIER     ::=     {id-ha 1}
id-sa-sqMod-nWithRSA       OBJECT IDENTIFIER     ::=     {id-sa 1}

END
```

附 件 G

认证通路约束使用举例

（本附件不是本建议书 | 国际标准的组成部分）

G.1 例子1：使用基本的约束

假设 Widget 公司想交叉认证 Acme 集团公司的中心 CA，但只想 Widget 团体使用由 CA 发放的终端实体证书，而不是由该 CA 认证的其他 CA 发放的证书。

Widget 公司可以通过为 Acme 集团公司的中心 CA 发放一个证书来满足该要求，包括以下扩展字段值：

基本约束字段的值：

{ cA TRUE, pathLenConstraint 0 }

G.2 例子2：使用策略映射和策略约束

假设在加拿大和美国政府之间需要进行以下交叉认证：

a） 某个加拿大政府的 CA 希望认证使用美国政府有关称为 *Can/US-Trade* 的加拿大政策的签名；

b） 美国政府有一项称为 *US/Can-Trade* 的政策，加拿大政府准备将之认为等同于它的 *Can/US-Trade* 政策；

c） 加拿大政府想采取保护措施，要求所有的美国证书明确声明支持该政策，并禁止映射至美国范围内的其他政策。

加拿大政府 CA 可以为美国政府 CA 发放一个带以下扩展字段值的证书：

证书策略字段的值：

{{ policyIdentifier -- 有关 *Can/US-Trade* 的对象标识符 -- **}}**

策略映射字段的值：

{{ issuerDomainPolicy -- 有关 *Can/US-Trade* 的对象标识符 --,

subjectDomainPolicy -- 有关 *US/Can-Trade* 的对象标识符 -- **}}**

PolicyConstraints 字段的值：

{{ policySet { -- 有关 *Can/US-Trade* 的对象标识符 -- **}, requireExplicitPolicy (0),**

inhibitPolicyMapping (0)}}

G.3 名称约束扩展的使用

G.3.1 带名称约束扩展的证书格式举例

通过在其 CA 证书中纳入名称约束扩展，CA 可以对其发放的证书以及在认证通路中的后续证书的对象名称（在 **subject** 字段或 **subjectAltName** 扩展中）施加各种不同限制。本节描述了带名称约束扩展的证书格式的例子。

为简化例子，在这些例子中，名称约束扩展所需的名称形式（**requiredNameForms**）只显示 rfc822 名称（**rfc822Name**）和 DN（**directoryName**）。

G.3.1.1 *permittedSubtrees*举例

（1-1） 如果 CA 证书包含以下名称约束扩展，那么对认证通路中的所有后续证书，DN 名称形式的每个对象名称（在 **subject** 字段或 **subjectAltName** 扩展中），如果它存在，那么应等同于或从属于美国的 Acme 公司（即{C=US, O=Acme Inc}）。

nameConstraints 扩展		
permittedSubtrees	**excludedSubtrees**	**requiredNameForms**
{{**base(directoryName**) {C=US, O=Acme Inc}}}	（无效）	（无效）

（1-2） 如果 CA 证书包含以下名称约束扩展，那么对认证通路中的所有后续证书，DN 名称形式的每个对象名称（在 **subject** 字段或 **subjectAltName** 扩展中），如果它存在，那么应等同于或直接从属于美国的 Acme 公司（即{C=US, O=Acme Inc}）。

nameConstraints 扩展		
permittedSubtrees	**excludedSubtrees**	**requiredNameForms**
{{**base(directoryName)** {C=US, O=Acme Inc}, **maximum 1**}}	（无效）	（无效）

（1-3） 如果 CA 证书包含以下名称约束扩展，那么对认证通路中的所有后续证书，DN 名称形式的每个对象名称（在 **subject** 字段或 **subjectAltName** 扩展中），如果它存在，那么应从属于美国的 Acme 公司（即{C=US, O=Acme Inc}）。

nameConstraints 扩展		
permittedSubtrees	**excludedSubtrees**	**requiredNameForms**
{{base(directoryName) {C=US, O=Acme Inc}, **minimum 1**}}	（无效）	（无效）

（1-4） 如果 CA 证书包含以下名称约束扩展，那么对认证通路中的所有后续证书，DN 名称形式的每个对象名称（在 **subject** 字段或 **subjectAltName** 扩展中），如果它存在，那么应等同于或从属于美国的 Acme 公司（即{C=US, O=Acme Inc}），或者等同于或从属于英国的 Acme 有限公司（{C=UK, O=Acme Ltd}）。

nameConstraints 扩展		
permittedSubtrees	**excludedSubtrees**	**requiredNameForms**
{{base(directoryName) {C=US, O=Acme Inc}}, **{base(directoryName)** {C=UK, O=Acme Ltd}}}	（无效）	（无效）

G.3.1.2 *excludedSubtrees*举例

（2-1） 如果 CA 证书包含以下名称约束扩展，那么对认证通路中的所有后续证书，DN 名称形式的每个对象名称（在 **subject** 字段或 **subjectAltName** 扩展中），如果它存在，那么不得等同于也不得从属于加拿大的 Acme 集团公司（即{C=CA, O=Acme Corp}）。

nameConstraints 扩展		
permittedSubtrees	**excludedSubtrees**	**requiredNameForms**
（无效）	**{{base(directoryName)** {C=CA, O=Acme Corp}}}	（无效）

（2-2） 如果 CA 证书包含以下名称约束扩展，那么对认证通路中的所有后续证书，DN 名称形式的每个对象名称（在 **subject** 字段或 **subjectAltName** 扩展中），如果它存在，那么不得从属于加拿大的 Acme 集团公司的各个直接下属（即{C=CA, O=Acme Corp}）。

nameConstraints 扩展		
permittedSubtrees	**excludedSubtrees**	**requiredNameForms**
（无效）	**{{base(directoryName)** {C=CA, O=Acme Corp}, **minimum 2**}}	（无效）

（2-3） 如果 CA 证书包含以下名称约束扩展，那么对认证通路中的所有后续证书，DN 名称形式的每个对象名称（在 **subject** 字段或 **subjectAltName** 扩展中），如果它存在，那么不得等同于加拿大的 Acme 集团公司（即{C=CA, O=Acme Corp}）。

nameConstraints 扩展		
permittedSubtrees	**excludedSubtrees**	**requiredNameForms**
（无效）	**{{base(directoryName)** {C=CA, O=Acme Corp}, **maximum 0**}}	（无效）

（2-4） 如果 CA 证书包含以下名称约束扩展，那么对认证通路中的所有后续证书，DN 名称形式的每个对象名称（在 **subject** 字段或 **subjectAltName** 扩展中），如果它存在，那么不得等同于也不得从属于加拿大的 Acme 集团公司（即{C=CA, O=Acme Corp}），并且不得等同于也不得从属于日本的 Asia Acme 公司（即{C=JP, O=Asia Acme}）。

nameConstraints 扩展		
permittedSubtrees	**excludedSubtrees**	**requiredNameForms**
（无效）	**{{base(directoryName)** {C=CA, O=Acme Corp}}, **{base(directoryName)** {C=JP, O=Asia Acme}}}	（无效）

G.3.1.3 permittedSubtrees和excludedSubtrees举例

（3-1） 如果 CA 证书包含以下名称约束扩展，那么对认证通路中的所有后续证书，DN 名称形式的每个对象名称（在 **subject** 字段或 **subjectAltName** 扩展中），如果它存在，那么应等同于或从属于美国的 Acme 公司（即{C=US, O=Acme Inc}），除了 Acme 公司的 R&D 机构单元和 R&D 机构的下属机构。

nameConstraints 扩展		
permittedSubtrees	**excludedSubtrees**	**requiredNameForms**
{{base(directoryName) {C=US, O=Acme Inc}}}	**{{base(directoryName)** {C=US, O=Acme Inc, OU=R&D}}}	（无效）

（3-2） 如果 CA 证书包含以下名称约束扩展，那么对认证通路中的所有后续证书，DN 名称形式的每个对象名称（在 **subject** 字段或 **subjectAltName** 扩展中），如果它存在，那么应等同于美国 Acme 公司的其中一个直接下属（即{C=US, O=Acme Inc}），除了采购机构单元（即{C=US, O=Acme Inc, OU=Purchasing}）。

nameConstraints 扩展		
permittedSubtrees	**excludedSubtrees**	**requiredNameForms**
{{base(directoryName) {C=US, O=Acme Inc}, **minimum 1, maximum 1**}}	**{{base(directoryName)** {C=US, O=Acme Inc, OU=Purchasing}}}	（无效）

G.3.1.4 带requiredNameForms的permittedSubtrees和excludedSubtrees举例

（4-1） 如果 CA 证书包含以下名称约束扩展，那么对认证通路中的所有后续证书，至少一个证书的对象名称（在 **subject** 字段或 **subjectAltName** 扩展中）应为 DN 名称形式。不过，每个对象名称都不受任何名称空间的约束。

nameConstraints 扩展			
permittedSubtrees	**excludedSubtrees**	**requiredNameForms**	
		rfc822 名称	DN
（无效）	（无效）	关	开

（4-2） 如果 CA 证书包含以下名称约束扩展，那么对认证通路中的所有后续证书，至少一个对象名称（在 **subject** 字段或 **subjectAltName** 扩展中）应为 DN 名称形式。另外，DN 名称形式的每个对象名称都应满足受 **permittedSubtrees** 和 **excludedSubtrees** 约束的名称空间的要求。

nameConstraints 扩展			
permittedSubtrees	**excludedSubtrees**	**requiredNameForms**	
		rfc822 名称	DN
{{base(directoryName) {C=JP, O=Asia Acme}}}	**{{base(directoryName)** {C=JP, O=Asia Acme, OU=Marketing}}}	关	开

（4-3） 如果 CA 证书包含以下名称约束扩展，那么对认证通路中的所有后续证书，DN 名称形式的每个对象名称（在 **subject** 字段或 **subjectAltName** 扩展中），如果存在，那么都应满足受 **permittedSubtrees** 和 **excludedSubtrees** 约束的名称空间的要求。

nameConstraints 扩展			
permittedSubtrees	**excludedSubtrees**	**requiredNameForms**	
		rfc822 名称	DN
{{base(directoryName) {C=JP, O=Asia Acme}}}	**{{base(directoryName)** {C=JP, O=Asia Acme, OU=Marketing}}}	关	关

注 — CA 证书的上述例子兼容于以下带名称约束扩展而不带 **requiredNameForms** 元素的 CA 证书。

nameConstraints 扩展		
permittedSubtrees	**excludedSubtrees**	**requiredNameForms**
{{base(directoryName) {C=JP, O=Asia Acme}}}	**{{base(directoryName)** {C=JP, O=Asia Acme, OU=Marketing}}}	（无效）

（4-4） 如果 CA 证书包含以下名称约束扩展，那么对认证通路中的所有后续证书，DN 名称形式的每个对象名称（在 **subject** 字段或 **subjectAltName** 扩展中），如果存在，那么都应满足受 **permittedSubtrees** 和 **excludedSubtrees** 约束的名称空间的要求。另外，至少一个 **rfc822Name** 名称形式的 **subjectAltName** 应出现，虽然其名称不受任何名称空间的约束。

nameConstraints 扩展			
permittedSubtrees	**excludedSubtrees**	**requiredNameForms**	
		rfc822 名称	DN
{{base(directoryName) {C=JP, O=Asia Acme}}}	**{{base(directoryName)** {C=JP, O=Asia Acme, OU=Marketing}}}	开	关

（4-5） 如果 CA 证书包含以下名称约束扩展，那么对认证通路中的所有后续证书，至少一个证书的对象名称（在 **subject** 字段或 **subjectAltName** 扩展中）应为 DN 名称形式或 rfc822 名称形式。DN 名称形式的每个对象名称，如果存在，那么都应满足受 **permittedSubtrees** 和 **excludedSubtrees** 约束的名称空间的要求。每个 **rfc822Name** 名称形式的对象名称都不受任何名称空间的约束。

nameConstraints 扩展			
permittedSubtrees	**ExcludedSubtrees**	**requiredNameForms**	
		rfc822 名称	DN
{{base(directoryName) {C=JP, O=Asia Acme}}}	**{{base(directoryName)** {C=JP, O=Asia Acme, OU=Marketing}}}	开	开

G.3.2 带名称约束扩展的证书处理举例

本节描述了在证书处理过程中如何利用通路处理状态变量来验证对象名称（在 **subject** 字段或 **subjectAltName** 扩展中）的例子，即 *permitted-subtrees*，*excluded-subtrees* 和 *required-name-forms*。

为简化例子，在这些例子中，通路处理状态变量 *required-name-forms* 只显示 rfc822 名称（**rfc822Name**）、DN（**directoryName**）和 URI（**uniformResourceIdentifier**）。

G.3.2.1 通过DN名称形式中*permitted-subtrees*实现的名称空间约束

在这种情况下，出现在所议证书中的、DN 名称形式的每个对象名称（在 **subject** 字段或 **subjectAltName** 扩展中）都应满足通路处理状态变量 *permitted-subtrees* 的约束要求。

（1-1） 出现一个允许的 DN 子树，要求 DN 为 *required-name-forms* 形式。

通路处理状态变量				
permitted-subtrees	*excluded-subtrees*	*required-name-forms*		
		rfc822	DN	URI
{{base(directoryName) {C=US, O=Acme Inc}}}	无	关	开	关

可接受的证书例子

1	**subject** = {C=US, O=Acme Inc, OU=Purchasing}
2	**subject** = {} **subjectAltName(directoryName)** = {C=US, O=Acme Inc, OU=Purchasing}
3	**subject** = {C=US, O=Acme Inc, OU=Purchasing} **subjectAltName (rfc822Name)** = manager@purchasing.acme.com
4	**subject** = {} **subjectAltName (directoryName)** = {C=US, O=Acme Inc, OU=Purchasing} **subjectAltName (rfc822Name)** = manager@purchasing.acme.com
5	**subject** = {C=US, O=Acme Inc, OU=Purchasing} **subjectAltName (directoryName)** = {C=US, O=Acme Inc, OU=Accounting}

不可接受的证书例子

1	**subject** = {C=US, O=*Acme Ltd*,OU=Purchasing}
2	**subject** = {} **subjectAltName (rfc822Name)** = manager@purchasing.acme.com 注 — *DN 丢失*。
3	**subject** = {C=US, O=Acme Inc, OU=Purchasing} **subjectAltName (directoryName)** = {C=US, O=*Acme Ltd*, OU=Purchasing}
4	**subject** = {C=US, O=*Acme Ltd*, OU=Purchasing} **subjectAltName (directoryName)** = {C=US, O=Acme Inc, OU=Purchasing}
5	**subject** = {C=US, O=*Acme Ltd*, OU=Purchasing} **subjectAltName (directoryName)** = {C=US, O=*Acme Ltd*, OU=Accounting}

（1-2） 出现两个允许的 DN 子树，要求 DN 为 *required-name-forms* 形式。

通路处理状态变量				
permitted-subtrees	*excluded-subtrees*	*required-name-forms*		
		rfc822	DN	URI
{{base(directoryName) {C=US, O=Acme Inc}}, **{base(directoryName)** {C=US, O=Acme Ltd}}}	无	关	开	关

可接受的证书例子

1	**subject** = {C=US, O=Acme Ltd, OU=Purchasing}
2	**subject** = {} **subjectAltName(directoryName)** = {C=US, O=Acme Ltd, OU=Purchasing}
3	**subject** = {C=US, O=Acme Ltd, OU=Purchasing} **subjectAltName(rfc822Name)** = manager@purchasing.acme-ltd.com
4	**subject** = {} **subjectAltName(directoryName)** = {C=US, O=Acme Inc, OU=Purchasing} **subjectAltName(directoryName)** = {C=US, O=Acme Ltd, OU=Purchasing} **subjectAltName(rfc822Name)** = manager@purchasing.acme.com
5	**subject** = {C=US, O=Acme Ltd, OU=Purchasing} **subjectAltName(directoryName)** = {C=US, O=Acme Ltd, OU= Accounting}

不可接受的证书例子

1	**subject** = {C=US, O=*Acme International*, OU=Accounting}
2	**subject** = {} **subjectAltName(rfc822Name)** = manager@purchasing.acme.com 注 — *DN 丢失*。
3	**subject** = {C=US, O=Acme Inc, OU=Purchasing} **subjectAltName(directoryName)** = {C=US, O=*Acme International*, OU=Accounting}
4	**subject** = {C=US, O=*Acme International*, OU=Accounting} **subjectAltName(directoryName)** = {C=US, O=Acme Inc, OU=Purchasing}
5	**subject** = {C=US, O=*Acme International*, OU=Accounting} **subjectAltName(directoryName)** = {C=US, O=*Acme Corp*, OU=Accounting}
6	**subject** = {} **subjectAltName(directoryName)** = {C=US, O=Acme Inc, OU=Purchasing} **subjectAltName(directoryName)** = {C=US, O=*Acme International*, OU=Accounting} **subjectAltName(rfc822Name)** = manager@purchasing.acme.com

（1-3） 出现一个允许的 DN 子树，*required-name-forms* 为空。

通路处理状态变量				
permitted-subtrees	*excluded-subtrees*	*required-name-forms*		
		rfc822	DN	URI
{{base(directoryName) {C=US, O=Acme Inc}}}	无	空		

可接受的证书例子

1	**subject** = {C=US, O=Acme Inc, OU=Purchasing}
2	**subject** = {} **subjectAltName(directoryName)** = {C=US, O=Acme Inc, OU=Purchasing}
3	**subject** = {C=US, O=Acme Inc, OU=Purchasing} **subjectAltName(rfc822Name)** = manager@purchasing.acme.com
4	**subject** = {} **subjectAltName(directoryName)** = {C=US, O=Acme Inc, OU=Purchasing} **subjectAltName(rfc822Name)** = manager@purchasing.acme.com
5	**subject** = {C=US, O=Acme Inc, OU=Purchasing} **subjectAltName(directoryName)** = {C=US, O=Acme Inc, OU=Accounting}
6	**subject** = {} **subjectAltName(rfc822Name)** = manager@purchasing.acme.com

不可接受的证书例子

1	**subject** = {C=US, O=*Acme Ltd*,OU=Purchasing}
2	**subject** = {C=US, O=Acme Inc, OU=Purchasing} **subjectAltName(directoryName)** = {C=US, O=*Acme Ltd*, OU=Accounting}
3	**subject** = {C=US, O=*Acme Ltd*, OU=Accounting} **subjectAltName(directoryName)** = {C=US, O=Acme Inc, OU=Purchasing}
4	**subject** = {C=US, O=*Acme Ltd*, OU=Accounting} **subjectAltName(directoryName)** = {C=US, O=*Acme Ltd*, OU=Purchasing}

G.3.2.2 通过DN名称形式中*excluded-subtrees*实现的名称空间约束

在这种情况下，出现在所议证书中的、DN 名称形式的每个对象名称（在 **subject** 字段或 **subjectAltName** 扩展中）都应满足通路处理状态变量 *excluded-subtrees* 的约束要求。

（2-1） 出现一个排除在外的 DN 子树，要求 DN 为 *required-name-forms* 形式。

通路处理状态变量				
permitted-subtrees	*excluded-subtrees*	*required-name-forms*		
		rfc822	DN	URI
无	**{{base(directoryName)** {C=US, O=Acme Ltd}}}	关	开	关

可接受的证书例子

1	**subject** = {C=US, O=Acme Inc, OU=Purchasing}
2	**subject** = {} **subjectAltName(directoryName)** = {C=US, O=Acme Inc, OU=Purchasing}
3	**subject** = {C=US, O=Acme Inc, OU=Purchasing} **subjectAltName(rfc822Name)** = manager@purchasing.acme.com
4	**subject** = {} **subjectAltName(directoryName)** = {C=US, O=Acme Inc, OU=Purchasing} **subjectAltName(rfc822Name)** = manager@purchasing.acme.com
5	**subject** = {C=US, O=Acme Inc, OU=Purchasing} **subjectAltName(directoryName)** = {C=US, O=Acme Inc, OU=Accounting}

不可接受的证书例子

1	**subject** = {C=US, O=*Acme Ltd*, OU=Purchasing}
2	**subject** = {} **subjectAltName(rfc822Name)** = manager@purchasing.acme.com 注 — *DN 丢失*。
3	**subject** = {C=US, O=Acme Inc, OU=Purchasing} **subjectAltName(directoryName)** = {C=US, O=*Acme Ltd*, OU=Accounting}

（2-2） 出现两个排除在外的 DN 子树，要求 DN 为 *required-name-forms* 形式。

通路处理状态变量				
permitted-subtrees	*excluded-subtrees*	*required-name-forms*		
		rfc822	DN	URI
无	**{{base(directoryName)** {C=US, O=Acme Inc}}, **{base(directoryName)** {C=US, O=Acme Ltd}}}	关	开	关

可接受的证书例子

1	**subject** = {C=US, O=Acme International, OU=Purchasing}
2	**subject** = {} **subjectAltName(directoryName)** = {C=US, O=Acme International, OU=Purchasing}
3	**subject** = {C=US, O=Acme International, OU=Purchasing} **subjectAltName(rfc822Name)** = purchasing@acme-international.com
4	**subject** = {} **subjectAltName(directoryName)** = {C=US, O=Acme International, OU=Purchasing} **subjectAltName(directoryName)** = {C=US, O=Acme N.Y, OU=Purchasing} **subjectAltName(rfc822Name)** = purchasing@acme-international.com

不可接受的证书例子

1	**subject** = {C=US, O=*Acme Inc*, OU=Purchasing}
2	**subject** = {C=US, O=*Acme Ltd*, OU=Purchasing}
3	**subject** = {} **subjectAltName(rfc822Name)** = purchasing@acme-international.com 注 — *DN* 丢失。
4	**subject** = {C=US, O=*Acme Inc*, OU=Purchasing} **subjectAltName(directoryName)** = {C=US, O=Acme International, OU=Accounting}
5	**subject** = {} **subjectAltName(directoryName)** = {C=US, O=*Acme Inc*, OU=Purchasing} **subjectAltName(directoryName)** = {C=US, O=Acme International, OU=Purchasing} **subjectAltName(rfc822Name)** = purchasing@acme-international.com

（2-3） 出现一个排除在外的 DN 子树，*required-name-forms* 为空。

通路处理状态变量				
permitted-subtrees	*excluded-subtrees*	*required-name-forms*		
		rfc822	DN	URI
无	**{{base(directoryName)** {C=US, O=Acme Inc}}}	空		

可接受的证书例子

1	**subject** = {C=US, O=Acme Ltd, OU=Purchasing}
2	**subject** = {} **subjectAltName(directoryName)** = {C=US, O=Acme Ltd, OU=Purchasing}
3	**subject** = {C=US, O=Acme Ltd, OU=Purchasing} **subjectAltName(rfc822Name)** = manager@purchasing.acme-ltd.com
4	**subject** = {} **subjectAltName(directoryName)** = {C=US, O=Acme Ltd, OU=Purchasing} **subjectAltName(rfc822Name)** = manager@purchasing.acme-ltd.com
5	**subject** = {C=US, O=Acme Ltd, OU=Purchasing} **subjectAltName(directoryName)** = {C=US, O=Acme Ltd, OU=Accounting}
6	**subject** = {} **subjectAltName(rfc822Name)** = manager@purchasing.acme-ltd.com

不可接受的证书例子

1	**subject** = {C=US, O=*Acme Inc*, OU=Purchasing}
2	**subject** = {C=US, O=*Acme Inc*, OU=Purchasing} **subjectAltName(directoryName)** = {C=US, O=Acme Ltd, OU=Accounting}

G.3.2.3 只通过required-name-forms实现的名称空间约束

（3-1） 要求 DN 为 *required-name-forms* 形式。

通路处理状态变量				
permitted-subtrees	*excluded-subtrees*	*required-name-forms*		
		rfc822	DN	URI
无	无	关	开	关

可接受的证书例子

1	**subject** = {C=US, O=Acme Ltd, OU=Purchasing}
2	**subject** = {} **subjectAltName(directoryName)** = {C=JP, O=Acme Inc, OU=Purchasing}
3	**subject** = {C=JP, O=Acme Ltd, OU=Purchasing} **subjectAltName(rfc822Name)** = manager@purchasing.acme-ltd.com
4	**subject** = {} **subjectAltName(directoryName)** = {C=US, O=Acme Ltd, OU=Purchasing} **subjectAltName(rfc822Name)** = manager@purchasing.acme-ltd.com
5	**subject** = {C=JP, O=Acme Ltd, OU=Purchasing} **subjectAltName(directoryName)** = {C=US, O=Acme Ltd, OU=Accounting}

不可接受的证书例子

1	**subject** = {} **subjectAltName(rfc822Name)** = manager@purchasing.acme-ltd.com 注 — *DN 丢失*。
2	**subject** = {} **subjectAltName(uniformResourceIdentifier)** = http://purchasing.www.acme-ltd.com 注 — *DN 丢失*。

（3-2） 要求 DN 或 **rfc822Name** 为 *required-name-forms* 形式。

通路处理状态变量				
permitted-subtrees	*excluded-subtrees*	*required-name-forms*		
		rfc822	DN	URI
无	无	开	开	关

可接受的证书例子

1	**subject** = {C=US, O=Acme Ltd, OU=Purchasing}
2	**subject** = {} **subjectAltName(directoryName)** = {C=JP, O=Acme Inc, OU=Purchasing}
3	**subject** = {} **subjectAltName(rfc822Name)** = manager@purchasing.acme-ltd.com
4	**subject** = {} **subjectAltName(directoryName)** = {C=US, O=Acme Ltd, OU=Purchasing} **subjectAltName(rfc822Name)** = manager@purchasing.acme-ltd.com
5	**subject** = {} **subjectAltName(rfc822Name)** = manager@purchasing.acme-ltd.com **subjectAltName(rfc822Name)** = purchasing@acme-ltd.com

不可接受的证书例子

1	**subject** = {} **subjectAltName(uniformResourceIdentifier)** = http://purchasing.www.acme-ltd.com 注 — *DN 和 rfc822 丢失*。
2	**subject** = {} **subjectAltName(dNSName)** = www.acme-ltd.com 注 — *DN 和 rfc822 丢失*。

G.3.2.4 通过多个名称形式中*permitted-subtrees*实现的名称空间约束

在这种情况下，出现在所议证书中的、DN 名称形式或 rfc822 名称形式的每个对象名称（在 **subject** 字段或 **subjectAltName** 扩展中）都应满足通路处理状态变量 *permitted-subtrees* 的约束要求。

（4-1） 出现一个允许的 DN 子树和另一个允许的 **rfc822Name** 子树。另外，要求 DN 为 *required-name-forms* 形式。

通路处理状态变量				
permitted-subtrees	*excluded-subtrees*	*required-name-forms*		
		rfc822	DN	URI
{{base(directoryName) {C=US, O=Acme Inc}}, **{base(rfc822Name)** .acme.com}}	无	关	开	关

可接受的证书例子

1	**subject** = {C=US, O=Acme Inc, OU=Purchasing}
2	**subject** = {C=US, O=Acme Inc, OU=Purchasing} **subjectAltName(rfc822Name)** = manager@purchasing.acme.com
3	**subject** = {C=US, O=Acme Inc, OU=Purchasing} **subjectAltName(directoryName)** = {C=US, O=Acme Inc, OU=Accounting}
4	**subject** = {C=US, O=Acme Inc, OU=Purchasing} **subjectAltName(uniformResourceIdentifier)** = http://purchasing.www.acme-inc.com

不可接受的证书例子

1	**subject** = {C=US, O=*Acme Ltd*, OU=Purchasing}
2	**subject** = {} **subjectAltName(rfc822Name)** = manager@purchasing.acme.com 注 — *DN 丢失*。
3	**subject** = {C=US, O=*Acme Ltd*, OU=Purchasing} **subjectAltName(rfc822Name)** = manager@purchasing.acme.com
4	**subject** = {C=US, O=Acme Inc, OU=Purchasing} **subjectAltName(rfc822Name)** = *manager@purchasing.acme-inc.com*
5	**subject** = {C=US, O=*Acme Ltd*, OU=Purchasing} **subjectAltName(rfc822Name)** = *manager@purchasing.acme-ltd.com*
6	**subject** = {} **subjectAltName(uniformResourceIdentifier)** = http://purchasing.www.acme-inc.com 注 — *DN 丢失*。

（4-2） 出现一个允许的 DN 子树和另一个允许的 **rfc822Name** 子树。另外，要求至少一个 DN 或 **rfc822Name** 为 *required-name-forms* 形式。

通路处理状态变量				
permitted-subtrees	*excluded-subtrees*	*required-name-forms*		
		rfc822	DN	URI
{{base(directoryName) {C=US, O=Acme Inc}}, **{base(rfc822Name)** .acme.com}}	无	开	开	关

可接受的证书例子

1	**subject** = {C=US, O=Acme Inc, OU=Purchasing}
2	**subject** = {} **subjectAltName(rfc822Name)** = manager@purchasing.acme.com
3	**subject** = {C=US, O=Acme Inc, OU=Purchasing} **subjectAltName(rfc822Name)** = manager@purchasing.acme.com
4	**subject** = {C=US, O=Acme Inc, OU=Purchasing} **subjectAltName(directoryName)** = { C=US, O=Acme Inc, OU=Accounting}
5	**subject** = {C=US, O=Acme Inc, OU=Purchasing} **subjectAltName(uniformResourceIdentifier)** = http://purchasing.www.acme-inc.com

不可接受的证书例子

1	**subject** = {C=US, O=*Acme Ltd*, OU=Purchasing}
2	**subject** = {} **subjectAltName(rfc822Name)** = *manager@purchasing.acme-inc.com*
3	**subject** = {C=US, O=*Acme Ltd*, OU=Purchasing} **subjectAltName(rfc822Name)** = manager@purchasing.acme.com
4	**subject** = {C=US, O=Acme Inc, OU=Purchasing} **subjectAltName(rfc822Name)** = *manager@purchasing.acme-inc.com*
5	**subject** = {C=US, O=*Acme Ltd*, OU=Purchasing} **subjectAltName(rfc822Name)** = *manager@purchasing.acme-ltd.com*
6	**subject** = {} **subjectAltName(uniformResourceIdentifier)** = http://purchasing.www.acme-inc.com 注 — *DN 和 rfc822 丢失*。

（4-3） 出现一个允许的 DN 子树和另一个允许的 **rfc822Name** 子树。不要求任何名称形式为 *required-name-forms*。

通路处理状态变量				
permitted-subtrees	*excluded-subtrees*	*required-name-forms*		
		rfc822	DN	URI
{{base(directoryName) {C=US, O=Acme Inc}}, **{base(rfc822Name)** .acme.com}}	无	空		

可接受的证书例子

1	**subject** = {C=US, O=Acme Inc, OU=Purchasing}
2	**subject** = {} **subjectAltName(rfc822Name)** = manager@purchasing.acme.com
3	**subject** = {C=US, O=Acme Inc, OU=Purchasing} **subjectAltName(rfc822Name)** = manager@purchasing.acme.com
4	**subject** = {} **subjectAltName(uniformResourceIdentifier)** = http://purchasing.www.acme.com
5	**subject** = {C=US, O=Acme Inc, OU=Purchasing} **subjectAltName(uniformResourceIdentifier)** = http://purchasing.www.acme.com

不可接受的证书例子

1	**subject** = {C=US, O=*Acme Ltd*, OU=Purchasing}
2	**subject** = {} **subjectAltName(rfc822Name)** = *manager@purchasing.acme-inc.com*
3	**subject** = {C=US, O=*Acme Ltd*, OU=Purchasing} **subjectAltName(rfc822Name)** = manager@purchasing.acme.com
4	**subject** = {C=US, O=Acme Inc, OU=Purchasing} **subjectAltName(rfc822Name)** = *manager@purchasing.acme-inc.com*
5	**subject** = {C=US, O=*Acme Ltd*, OU=Purchasing} **subjectAltName(rfc822Name)** = *manager@purchasing.acme-inc.com*

G.3.2.5 通过多个名称形式中*excluded-subtrees*实现的名称空间约束

在这种情况下，出现在所议证书中的、DN 名称形式或 rfc822 名称形式的每个对象名称（在 **subject** 字段或 **subjectAltName** 扩展中）都应满足通路处理状态变量 *excluded-subtrees* 的约束要求。

（5-1） 出现一个排除在外的 DN 子树和另一个排除在外的 **rfc822Name** 子树。另外，要求 DN 为 *required-name-forms* 形式。

通路处理状态变量				
permitted-subtrees	*excluded-subtrees*	*required-name-forms*		
		rfc822	DN	URI
无	**{{base(directoryName)** {C=US, O=Acme Inc}}, **{base(rfc822Name)** .acme.com}}	关	开	关

可接受的证书例子

1	**subject** = {C=US, O=Acme Ltd, OU=Purchasing}
2	**subject** = {C=US, O=Acme Ltd, OU=Purchasing} **subjectAltName(rfc822Name)** = manager@purchasing.acme-ltd.com
3	**subject** = {C=US, O=Acme Ltd, OU=Purchasing} **subjectAltName(directoryName)** = {C=US, O=Acme Ltd, OU=Accounting}
4	**subject** = {C=US, O=Acme Ltd, OU=Purchasing} **subjectAltName(uniformResourceIdentifier)** = http://purchasing.www.acme-ltd.com

不可接受的证书例子

1	**subject** = {C=US, O=*Acme Inc*, OU=Purchasing}
2	**subject** = {} **subjectAltName(rfc822Name)** = manager@purchasing.acme.com 注 — *DN 丢失*。
3	**subject** = {C=US, O=Acme Ltd, OU=Purchasing} **subjectAltName(rfc822Name)** = *manager@purchasing.acme.com*
4	**subject** = {C=US, O=*Acme Inc*, OU=Purchasing} **subjectAltName(rfc822Name)** = manager@purchasing.acme-inc.com
5	**subject** = {C=US, O=*Acme Inc*, OU=Purchasing} **subjectAltName(rfc822Name)** = *manager@purchasing.acme.com*
6	**subject** = {C=US, O=Acme Ltd, OU=Purchasing} **subjectAltName(directoryName)** = {C=US, O=*Acme Inc*, OU=Accounting}
7	**subject** = {} **subjectAltName(uniformResourceIdentifier)** = http://purchasing.www.acme-inc.com 注 — *DN 丢失*。

（5-2） 出现一个排除在外的 DN 子树和另一个排除在外的 **rfc822Name** 子树。另外，要求至少一个 DN 或 **rfc822Name** 为 *required-name-forms* 形式。

通路处理状态变量				
permitted-subtrees	*excluded-subtrees*	*required-name-forms*		
		rfc822	DN	URI
无	**{{base(directoryName)** {C=US, O=Acme Inc}}, **{base(rfc822Name)** .acme.com}}	开	开	关

可接受的证书例子

1	**subject** = {C=US, O=Acme Ltd, OU=Purchasing}
2	**subject** = {} **subjectAltName(rfc822Name)** = manager@purchasing.acme.org
3	**subject** = {C=US, O=Acme Ltd, OU=Purchasing} **subjectAltName(rfc822Name)** = manager@purchasing.acme-ltd.com
4	**subject** = {C=US, O=Acme Ltd, OU=Purchasing} **subjectAltName(directoryName)** = {C=US, O=Acme Ltd, OU=Accounting}
5	**subject** = {C=US, O=Acme Ltd, OU=Purchasing} **subjectAltName(uniformResourceIdentifier)** = http://purchasing.www.acme-ltd.com

不可接受的证书例子

1	**subject** = {C=US, O=*Acme Inc*, OU=Purchasing}
2	**subject** = {} **subjectAltName(rfc822Name)** = *manager@purchasing.acme.com*
3	**subject** = {C=US, O=Acme Ltd, OU=Purchasing} **subjectAltName(rfc822Name)** = *manager@purchasing.acme.com*
4	**subject** = {C=US, O=*Acme Inc*, OU=Purchasing} **subjectAltName(rfc822Name)** = manager@purchasing.acme-inc.com
5	**subject** = {C=US, O=*Acme Inc*, OU=Purchasing} **subjectAltName(rfc822Name)** = *manager@purchasing.acme.com*
6	**subject =** {C=US, O=Acme Ltd, OU=Purchasing} **subjectAltName(directoryName)** = {C=US, O=*Acme Inc*, OU=Accounting}
7	**subject** = {} **subjectAltName(uniformResourceIdentifier)** = http://purchasing.www.acme-inc.com 注 — *DN 和 rfc822 丢失*。

（5-3） 出现一个排除在外的 DN 子树和另一个排除在外的 **rfc822Name** 子树。不要求任何名称形式为 *required-name-forms*。

通路处理状态变量				
permitted-subtrees	*excluded-subtrees*	*required-name-forms*		
		rfc822	DN	URI
无	**{{base(directoryName)** {C=US, O=Acme Inc}}, **{base(rfc822Name)** .acme.com}}	空		

可接受的证书例子

1	**subject** = {C=US, O=Acme Ltd, OU=Purchasing}
2	**subject** = {} **subjectAltName(rfc822Name)** = manager@purchasing.acme-ltd.com
3	**subject** = {C=US, O=Acme Ltd, OU=Purchasing} **subjectAltName(rfc822Name)** = manager@purchasing.acme-ltd.com
4	**subject** = {} **subjectAltName(uniformResourceIdentifier)** = http://purchasing.www.acme-inc.com
5	**subject** = {C=US, O=Acme Ltd, OU=Purchasing} **subjectAltName(uniformResourceIdentifier)** = http://purchasing.www.acme-ltd.com

不可接受的证书例子

1	**subject** = {C=US, O=*Acme Inc*, OU=Purchasing}
2	**subject** = {} **subjectAltName(rfc822Name)** = *manager@purchasing.acme.com*
3	**subject** = {C=US, O=Acme Ltd, OU=Purchasing} **subjectAltName(rfc822Name)** = *manager@purchasing.acme.com*
4	**subject** = {C=US, O=*Acme Inc*, OU=Purchasing} **subjectAltName(rfc822Name)** = manager@purchasing.acme-inc.com
5	**subject** = {C=US, O=*Acme Inc*, OU=Purchasing} **subjectAltName(rfc822Name)** = *manager@purchasing.acme.com*
6	**subject** = {C=US, O=Acme Ltd, OU=Purchasing} **subjectAltName(directoryName)** = {C=US, O=*Acme Inc*, OU=Accounting}

附 件 H

确定认证通路对哪个策略有效的指南

（本附件不是本建议书 | 国际标准的组成部分）

本附件的目的是为具有 PKI 功能的应用提供有关证书通路验证的证书策略相关处理控制的指南。PKI 通过证书内容进行的证书策略相关处理控制在本规范有关证书通路处理程序的节中描述。

本附件对通路处理程序中两个策略相关输入的初始化进行了阐述：*initial-policy-set* 和 *initial-explicit-policy*。除了这些之外，也可以通过用户初始化的 *initial-policy-mapping-inhibit* and *initial-inhibit-any-policy* inputs to the procedure，在通路处理期间，对策略相关信息的处理也有影响；不过，这些不在本附件的范畴之内。将 *initial-policy-mapping-inhibit* 设为 **TRUE** 将防止策略映射用于成功的通路验证中。将 *initial-inhibit-any-policy* 设为 **TRUE** 将防止 **anyPolicy** 的特殊 OID（如果在证书出现的话）是一个有关某个特定策略 OID 的可接受的匹配。

本附件中的术语“用户”可以是一个“作为人的用户”，或者是一个具有 PKI 功能的“应用”。

假设以下情况：

1） 用户要求认证通路对用户感兴趣的其中一个策略是有效的。

2） 用户要求认证通路对至少一个策略是有效的，但用户不必关心到底是哪个策略。当用户计划利用其他的上下文信息和信息内容进行额外的策略处理时，应该（可以）使用这种假设的情况，以确定用户是否接受认证通路为有效的其中一个策略，以便用于特殊的事务处理。

3） 对认证通路，用户没有任何策略相关的要求。换句话说，用户希望接受一条对任何策略都无效的认证通路，否则是有效的。

4） 用户希望认证通路对用户感兴趣的其中一个策略是有效的，但如果不是这样，则希望能有机会对用户感兴趣的策略无效的通路重新进行考虑。当用户通常要求认证通路对用户可接受的策略为有效时，应该（可以）使用这种假设的情况，但应基于其他上下文信息和信息内容，用户可以希望不管策略失败与否。

以下条目描述了用户如何从一个相应的通路验证机制中获得期望的信息。

H.1 对要求的用户特定的策略有效的认证通路

在这种情况下，用户要求认证通路对用户感兴趣的其中一个策略是有效的。为了获得期望的信息，用户应将策略处理相关的认证通路验证输入设置如下：

initial-policy-set = {set of policies of interest to the user}

initial-explicit-policy = **TRUE**

如果通路验证成功，那么认证通路对用户感兴趣的至少一个策略是有效的。认证通路对 *user-constrained-policy-set* 输出变量中所列的各策略是有效的。

在这种情况下，如果通路验证机制拒绝它，那么应用不得将认证通路用于证书策略相关的失败 [4)]。

4） 当故障是由证书策略相关的扩展或证书策略相关的状态变量引起时，则通路验证故障为证书策略相关的故障。证书策略相关的扩展为：**certificatePolicies**、**policyMappings**、**policyConstraints**和**inhibitAnyPolicy**。证书策略相关的状态变量为：*authorities-constrained-policy-set*、*explicit-policy-indicator*、*policy-mapping-inhibit-indicator*和*inhibit-any-policy-indicator*。

H.2 对任何要求的策略都有效的认证通路

在这种情况下，用户要求认证通路对至少一个策略是有效的，但用户不必关心到底是哪个策略。为了获得期望的信息，用户应将策略处理相关的认证通路验证输入设置如下：

initial-policy-set = {**anyPolicy**}

initial-explicit-policy = **TRUE**

如果通路验证成功，那么认证通路对至少一个策略是有效的。认证通路对 *user-constrained-policy-set* 输出变量中所列的各策略是有效的。

在这种情况下，如果通路验证机制拒绝它，那么应用不得将认证通路用于证书策略相关的失败。

H.3 无论策略如何都有效的认证通路

在这种情况下，用户对认证通路没有任何策略相关的要求。为了获得期望的信息，用户应将策略处理相关的认证通路验证输入设置如下：

initial-policy-set = {**anyPolicy**}

initial-explicit-policy = **FALSE**

如果通路验证成功，那么认证通路对 *user-constrained-policy-set* 输出变量中所列的各策略是有效的。

在这种情况下，如果通路验证机制拒绝它，那么应用不得将认证通路用于证书策略相关的失败。

应注意：在这种情况下，认证通路可以出现策略相关的失败。一个例子是：如果基础设施（即认证通路中的一个 CA 证书）引起 *explicit-policy-indicator* 设置。在这种情况下，如果通路对任何策略都无效，即 *authorities-constrained-policy-set* 为空，那么相应的通路验证机制将返回一个失败。应用应对这种类型的失败拒绝认证通路。

H.4 对希望的而非要求的用户特定的策略有效的认证通路

在这种情况下，用户要求认证通路对用户感兴趣的其中一个策略是有效的，当不希望拒绝对用户感兴趣的任何策略都无效的通路。为了获得期望的信息，用户应将策略处理相关的认证通路验证输入设置如下：

initial-policy-set = {set of policies of interest to the user}

initial-explicit-policy = **FALSE**

如果通路验证成功，那么认证通路对 *user-constrained-policy-set* 输出变量中所列的各策略是有效的。*user-constrained-policy-set* 是 *initial-policy-set* 的一个子集。请注意：在这种情况下，当未设置 *explicit-policy-indicator* 时，*user-constrained-policy-set* 可以为 **NULL**。应用应对返回的 *user-constrained-policy-set* 进行检查，以确定用户是否接受通路。

在这种情况下，应用应对基础设施引起的策略相关的失败拒绝认证通路，（即当 *authorities-constrained-policy-set* 为空且设置了 *explicit-policy-indicator* 时）。

应注意：在这种情况下，认证通路可以出现策略相关的失败。一个例子是：如果基础设施（即认证通路中的一个 CA 证书）引起 *explicit-policy-indicator* 设置。在这种情况下，如果通路对任何策略都无效，即 *authorities-constrained-policy-set* 为空，那么相应的通路验证机制将返回一个失败。应用应对这种类型的失败拒绝认证通路。

另一个例子是：结合用户输入，并且基础设施引起策略相关的失败。这在认证通路中的 CA 证书引起 *explicit-policy-indicator* 设置、*authorities-constrained-policy-set* 非空、*user-constrained-policy-set* 为空时发生。相应的通路验证机制将返回一个失败。在这些条件下，如果通路验证机制返回一个失败的惟一理由是 *user-constrained-policy-set* 为空，那么应用可以选择不管失败与否而接受认证通路。由于 *authorities-constrained-policy-set* 不为空，因此需继续关注受机构影响的约束。应用接受本通路等同于应用重新向验证机制提供通路，此时 *initial-policy-set* 等于 **anyPolicy**，*initial-explicit-policy* 等于 **FALSE**，并且需要对返回的 *user-constrained-policy-set* 进行检查，以确定是否接受通路。

附 件 I

密钥用法证书扩展问题

（本附件不是本建议书 | 国际标准的组成部分）

依据使用证书的安全环境，keyUsage 证书扩展中 contentCommitment 位与其他 keyUsage 位的结合可能具有安全含意。如果对象所处的环境是完全受控的和可信任的，那么没有特定的安全含意。例如，在对象完全可信任的情况下，即数据经准确签署，或者使用了鉴权协议的安全特性。如果对象所处的环境不是完全受控的或者不是完全可信任的，那么无意间签署承诺是可能的。例子包括使用以不良方式形成的鉴权交换、使用不可靠的软件部件。如果对象使用了不可信任的环境，那么可以通过采取以下措施来限制这些安全含意：

— 不将证书中的 contentCommitment 密钥用法设置与任何其他密钥用法设置相结合，对该证书只使用相应的专用密钥；

— 限制将与拥有 contentCommitment 密钥用法位设置的证书相关的专用密钥用于认为是完全受控的和可信任的环境。

附 件 J

按字母顺序的信息项目定义清单

（本附件不是本建议书 | 国际标准的组成部分）

本附件为在本号码簿规范中定义的证书和 CRL 格式定义、证书扩展、对象类别、名称形式、属性类型以及匹配规则等提供了一个按字母顺序的索引。

项　　目	节
证书和 ***CRL*** 格式	
属性证书格式	12.1
属性撤消清单	7.3
公开密钥证书格式	7
证书、***CRL*** 和 ***CRL*** 条目扩展	
可接受的证书策略扩展	15.5.2.3
可接受的特权策略扩展	15.1.2.4
属性描述符扩展	15.3.2.2
机构属性标识符扩展	15.5.2.4
机构密钥标识符扩展	8.2.2.1
基础更新扩展	8.6.2.5
基本的属性约束扩展	15.5.2.1
基本的约束扩展	8.4.2.1
证书发放者扩展	8.6.2.3
证书策略扩展	8.2.2.6
CRL 分发点扩展	8.6.2.1
CRL 号码扩展	8.5.2.1
CRL 范围扩展	8.5.2.5
CRL 流标识符扩展	8.5.2.7
委托名称约束扩展	15.5.2.2
Delta CRL 指示符扩展	8.6.2.4
Delta 信息扩展	8.5.2.9
有关 CRL 扩展的过期证书	8.5.2.12
经扩展的密钥用法扩展	8.2.2.4
最新的 CRL 扩展	8.6.2.6
持有指令代码扩展	8.5.2.3
间接的发放者扩展	15.1.2.5
禁止任何策略扩展	8.4.2.4
无效日期扩展	8.5.2.4
代表扩展发布	15.5.2.6
发放者可选的名称扩展	8.3.2.2
发放分发点扩展	8.6.2.2
密钥用法扩展	8.2.2.3
名称约束扩展	8.4.2.2
无声明扩展	15.1.2.6
无撤消信息扩展	15.2.2.2
按序的清单扩展	8.5.2.8
策略约束扩展	8.4.2.3
策略映射扩展	8.2.2.7
专用密钥用法期限扩展	8.2.2.5
理由代码扩展	8.5.2.2

项　　目	节
属性证书准确匹配	17.3.1
属性证书匹配	17.3.2
属性标识符匹配	15.3.2.2.1
基本的属性约束匹配	15.5.2.1.1
证书准确匹配	11.3.1
证书清单准确匹配	11.3.5
证书清单匹配	11.3.6
证书匹配	11.3.2
证书对准确匹配	11.3.3
证书对匹配	11.3.4
委托的名称约束匹配	15.5.2.2.1
委托通路匹配	17.3.4
增强的证书匹配	11.3.10
持有者发放者匹配	17.3.3
间接的发放者匹配	15.1.2.5
PKI 通路匹配	11.3.9
策略匹配	11.3.8
角色规范证书 ID 匹配	15.4.2.1.1
SOA 标识符匹配	15.3.2.1.1
时间规范匹配	15.1.2.1.1

附 件 K

修正和勘误表

（本附件不是本建议书 | 国际标准的组成部分）

本号码簿规范的本版本包括以下修正草案，它们经 ISO/IEC 投票批准。

— 公开密钥和属性证书扩展修正 4。

本号码簿规范的本版本包括技术上的勘误，用于依照本规范的第 4 版本，纠正下述的缺陷报告：

— 技术勘误表 1（包括缺陷报告 272、273、274、275、276、277、278 和 279）；

— 技术勘误表 2（包括缺陷报告 284、285 和 286）；以及

— 技术勘误表 3（包括缺陷报告 281、282、289、291、296、298、299、300、301、304 和 305）。

ITU-T 系列建议书

A系列	ITU-T工作的组织
D系列	一般资费原则
E系列	综合网络运行、电话业务、业务运行和人为因素
F系列	非话电信业务
G系列	传输系统和媒质、数字系统和网络
H系列	视听及多媒体系统
I系列	综合业务数字网
J系列	有线网络和电视、声音节目及其它多媒体信号的传输
K系列	干扰的防护
L系列	电缆和外部设备其它组件的结构、安装和保护
M系列	电信管理，包括TMN和网络维护
N系列	维护：国际声音节目和电视传输电路
O系列	测量设备的技术规范
P系列	电话传输质量、电话设施及本地线路网络
Q系列	交换和信令
R系列	电报传输
S系列	电报业务终端设备
T系列	远程信息处理业务的终端设备
U系列	电报交换
V系列	电话网上的数据通信
X系列	**数据网、开放系统通信和安全性**
Y系列	全球信息基础设施、互联网协议问题和下一代网络
Z系列	用于电信系统的语言和一般软件问题

瑞士印刷
2006年，日内瓦

页 4: [1] 批注 [P1] POOL

Page: 4
T: **3.3.1 attribute certificate (AC)**

页 4: [2] 批注 [P2] POOL

Page: 4
D: A data structure, digitally signed by an Attribute Authority, that binds some attribute values with identification information about its holder.

页 4: [3] 批注 [P3] POOL

Page: 4
T: **3.3.2 Attribute Authority (AA)**

页 4: [4] 批注 [P4] POOL

Page: 4
D: An authority which assigns privileges by issuing attribute certificates.

页 4: [5] 批注 [P5] POOL

Page: 4
T: **3.3.3 attribute authority revocation list (AARL)**

页 4: [6] 批注 [P6] POOL

Page: 4
D: A revocation list containing a list of references to attribute certificates issued to AAs that are no longer considered valid by the issuing authority.

页 4: [7] 批注 [P7] POOL

Page: 4
T: **3.3.4 attribute certificate revocation list (ACRL)**

页 4: [8] 批注 [P8] POOL

Page: 4
D: A revocation list containing a list of references to attribute certificates that are no longer considered valid by the issuing authority.

页 4: [9] 批注 [P9] POOL

Page: 4
T: **3.3.5 authentication token; (token)**

页 4: [10] 批注 [P10] POOL

Page: 4
D: Information conveyed during a strong authentication exchange, which can be used to authenticate its sender.

页 4: [11] 批注 [P11] POOL

Page: 4
T: **3.3.6 authority**

页 4: [12] 批注 [P12] POOL

Page: 4
D: An entity, responsible for the issuance of certificates. Two types are defined in this Specification; certification authority which issues public-key certificates and attribute authority which issues attribute certificates.

页 4: [13] 批注 [P13] POOL

Page: 4
T: **3.3.7 authority certificate**

页 4: [14] 批注 [P14] POOL

Page: 4
D: A certificate issued to an authority (e.g., either to a certification authority or to an attribute authority).

页 4: [15] 批注 [P15] POOL

Page: 4
T: **3.3.8 base CRL**

页 4: [16] 批注 [P16] POOL

Page: 4
D: A CRL that is used as the foundation in the generation of a dCRL.

页 4: [17] 批注 [P17] POOL

Page: 4
T: **3.3.9 CA-certificate**

页 4: [18] 批注 [P18] POOL

Page: 4
D: A certificate for one CA issued by another CA.

页 4: [19] 批注 [P19] POOL

Page: 4
T: **3.3.10 certificate policy**

页 4: [20] 批注 [P20] POOL

Page: 4
D: A named set of rules that indicates the applicability of a certificate to a particular community and/or class of application with common security requirements. For example, a particular certificate policy might indicate applicability of a type of certificate to the authentication of electronic data interchange transactions for the trading of goods within a given price range.

页 4: [21] 批注 [P21] POOL

Page: 4
T: **3.3.11 certification practice statement (CPS**

页 4: [22] 批注 [P22] POOL

Page: 4
A: **certification practice statement (CPS)**

页 4: [23] 批注 [P23] POOL

Page: 4
D: A statement of the practices that a Certification Authority employs in issuing certificates.

页 4: [24] 批注 [P24] POOL

Page: 4
T: **3.3.12 certificate revocation list (CRL)**

页 4: [25] 批注 [P25] POOL

Page: 4
D: A signed list indicating a set of certificates that are no longer considered valid by the certificate issuer. In addition to the generic term CRL, some specific CRL types are defined for CRL that cover particular scopes.

页 4: [26] 批注 [P26] POOL

Page: 4
T: **3.3.13 certificate user**

页 4: [27] 批注 [P27] POOL

Page: 4
D: An entity that needs to know, with certainty, the attributes and/or public key of another entity.

页 4: [28] 批注 [P28] POOL

Page: 4
T: **3.3.14 certificate serial number**

页 4: [29] 批注 [P29] POOL

Page: 4
D: An integer value, unique within the issuing authority, which is unambiguously associated with a certificate issued by that authority.

页 4: [30] 批注 [P30] POOL

Page: 4
T: **3.3.15 certificate-using system**

页 4: [31] 批注 [P31] POOL

Page: 4
D: An implementation of those functions defined in this Directory Specification that are used by a certificate-user.

页 4: [32] 批注 [P32] POOL

Page: 4
T: **3.3.16 certificate validation**

页 4: [33] 批注 [P33] POOL

Page: 4
D: The process of ensuring that a certificate was valid at a given time, including possibly the construction and processing of a certification path, and ensuring that all certificates in that path were valid (i.e., were not expired or revoked) at that given time.

页 4: [34] 批注 [P34] POOL

Page: 4
T: **3.3.17 certification authority (CA)**

页 4: [35] 批注 [P35] POOL

Page: 4
D: An authority trusted by one or more users to create and assign public-key certificates. Optionally the certification authority may create the users' keys.

页 4: [36] 批注 [P36] POOL

Page: 4
T: **3.3.18 certification authority revocation list (CARL)**

页 4: [37] 批注 [P37] POOL

Page: 4
D: A revocation list containing a list of public-key certificates issued to certification authorities, that are no longer considered valid by the certificate issuer.

页 4: [38] 批注 [P38] POOL

Page: 4
T: **3.3.19 certification path**

页 4: [39] 批注 [P39] POOL

Page: 4
D: An ordered sequence of public key certificates of objects in the DIT which, together with the public key of the initial object in the path, can be processed to obtain that of the final object in the path.

页 4: [40] 批注 [P40] POOL

Page: 4
T: **3.3.20 CRL distribution point**

页 4: [41] 批注 [P41] POOL

Page: 4
D: A directory entry or other distribution source for CRL; a CRL distributed through a CRL distribution point may contain revocation entries for only a subset of the full set of certificates issued by one CA or may contain revocation entries for multiple CA.

页 4: [42] 批注 [P42] POOL

Page: 4
T: **3.3.21 cross-certificate**

页 4: [43] 批注 [P43] POOL

Page: 4
D: A public-key or attribute certificate where the issuer and the subject/holder are different CA or AAs respectively. CA and AAs issue cross-certificates to other CA or AAs respectively as a mechanism to authorize the subject CA's existence (e.g., in a strict hierarchy) or to recognize the existence of the subject CA or holder AA (e.g., in a distributed trust model). The cross-certificate structure is used for both of these.

页 5: [44] 批注 [P44] POOL

Page: 5
T: **3.3.22 cryptographic system, cryptosystem**

页 5: [45] 批注 [P45] POOL

Page: 5
D: A collection of transformations from plain text into ciphertext and vice versa, the particular transformation(s) to be used being selected by keys. The transformations are normally defined by a mathematical algorithm.

页 5: [46] 批注 [P46] POOL

Page: 5
T: **3.3.23 data confidentiality**

页 5: [47] 批注 [P47] POOL

Page: 5
D: This service can be used to provide for protection of data from unauthorized disclosure. The data confidentiality service is supported by the authentication framework. It can be used to protect against data interception.

页 5: [48] 批注 [P48] POOL

Page: 5
T: **3.3.24 delegation**

页 5: [49] 批注 [P49] POOL

Page: 5
D: Conveyance of privilege from one entity that holds such privilege, to another entity.

页 5: [50] 批注 [P50] POOL

Page: 5
T: **3.3.25 delegation path**

页 5: [51] 批注 [P51] POOL

Page: 5
D: An ordered sequence of certificates which, together with authentication of a privilege asserter's identity can be processed to verify the authenticity of an asserter's privilege

页 5: [52] 批注 [P52] POOL

Page: 5
T: **3.3.26 delta-CRL (dCRL)**

页 5: [53] 批注 [P53] POOL

Page: 5
D: A partial revocation list that only contains entries for certificates that have had their revocation status changed since the issuance of the referenced base CRL.

页 5: [54] 批注 [P54] POOL

Page: 5
T: **3.3.27 end entity**

页 5: [55] 批注 [P55] POOL

Page: 5
D: Either a public key certificate subject that uses its private key for purposes other than signing certificates, or an attribute certificate holder that uses its attributes to gain access to a resource, or an entity that is a relying party.

页 5: [56] 批注 [P56] POOL

Page: 5
T: **3.3.28 end-entity attribute certificate revocation list (EARL)**

页 5: [57] 批注 [P57] POOL

Page: 5
D: A revocation list containing a list of attribute certificates issued to holders, that are not also AAs, that are no longer considered valid by the certificate issuer.

页 5: [58] 批注 [P58] POOL

Page: 5
T: **3.3.29 end-entity public-key certificate revocation list (EPRL)**

页 5: [59] 批注 [P59] POOL

Page: 5
D: A revocation list containing a list of public-key certificates issued to subjects, that are not also CA, that are no longer considered valid by the certificate issuer.

页 5: [60] 批注 [P60] POOL

Page: 5
T: **3.3.30 environmental variables**

页 5: [61] 批注 [P61] POOL

Page: 5
D: Those aspects of policy required for an authorization decision, that are not contained within static structures, but are available through some local means to a privilege verifier (e.g., time of day or current account balance).

页 5: [62] 批注 [P62] POOL

Page: 5
T: **3.3.31 full CRL**

页 5: [63] 批注 [P63] POOL

Page: 5
D: A complete revocation list that contains entries for all certificates that have been revoked for the given scope.

页 5: [64] 批注 [P64] POOL

Page: 5
T: **3.3.32 hash function**

页 5: [65] 批注 [P65] POOL

Page: 5
D: A (mathematical) function which maps values from a large (possibly very large) domain into a smaller range. A "good" hash function is such that the results of applying the function to a (large) set of values in the domain will be evenly distributed (and apparently at random) over the range.

页 5: [66] 批注 [P66] POOL

Page: 5
T: **3.3.33 holder**

页 5: [67] 批注 [P67] POOL

Page: 5
D: An entity to whom some privilege has been delegated either directly from the Source of Authority or indirectly through another Attribute Authority.

页 5: [68] 批注 [P68] POOL

Page: 5
T: **3.3.34 indirect CRL (iCRL)**

页 5: [69] 批注 [P69] POOL

Page: 5
D: A revocation list that at least contains revocation information about certificates issued by authorities other than that which issued this CRL.

页 5: [70] 批注 [P70] POOL

Page: 5
T: **3.3.35 key agreement**

页 5: [71] 批注 [P71] POOL

Page: 5
D: A method for negotiating a key value on-line without transferring the key, even in an encrypted form, e.g., the Diffie-Hellman technique (see ISO/IEC 11770-1 for more information on key agreement mechanisms).

页 5: [72] 批注 [P72] POOL

Page: 5
T: **3.3.36 object method**

页 5: [73] 批注 [P73] POOL

Page: 5
D: An action that can be invoked on a resource (e.g., a file system may have read, write and execute object methods).

页 5: [74] 批注 [P74] POOL

Page: 5
T: **3.3.37 one-way function**

页 5: [75] 批注 [P75] POOL

Page: 5
D: A (mathematical) function f which is easy to compute, but which for a general value y in the range, it is computationally difficult to find a value x in the domain such that f(x) = y. There may be a few values y for which finding x is not computationally difficult.

页 5: [76] 批注 [P76] POOL

Page: 5
T: **3.3.38 policy mapping**

页 5: [77] 批注 [P77] POOL

Page: 5
D: Recognizing that, when a CA in one domain certifies a CA in another domain, a particular certificate policy in the second domain may be considered by the authority of the first domain to be equivalent (but not necessarily identical in all respects) to a particular certificate policy in the first domain.

页 5: [78] 批注 [P78] POOL

Page: 5
T: **3.3.39 private key; secret key** (deprecated)

页 5: [79] 批注 [P79] POOL

Page: 5
D: (In a public key cryptosystem) that key of a user's key pair which is known only by that user.

页 5: [80] 批注 [P80] POOL

Page: 5
T: **3.3.40 privilege**

页 5: [81] 批注 [P81] POOL

Page: 5
D: An attribute or property assigned to an entity by an authority.

页 5: [82] 批注 [P82] POOL

Page: 5
T: **3.3.41 privilege asserter**

页 5: [83] 批注 [P83] POOL

Page: 5
D: A privilege holder using their attribute certificate or public-key certificate to assert privilege.

页 5: [84] 批注 [P84] POOL

Page: 5
T: **3.3.42 privilege management infrastructure (PMI)**

页 5: [85] 批注 [P85] POOL

Page: 5
D: The infrastructure able to support the management of privileges in support of a comprehensive authorization service and in relationship with a Public Key Infrastructure.

页 6: [86] 批注 [P86] POOL

Page: 6
T: **3.3.43 privilege policy**

页 6: [87] 批注 [P87] POOL

Page: 6
D: The policy that outlines conditions for privilege verifiers to provide/perform sensitive services to/for qualified privilege asserters. Privilege policy relates attributes associated with the service as well as attributes associated with privilege asserters.

页 6: [88] 批注 [P88] POOL

Page: 6
T: **3.3.44 privilege verifier**

页 6: [89] 批注 [P89] POOL

Page: 6
D: An entity verifying certificates against a privilege policy.

页 6: [90] 批注 [P90] POOL

Page: 6
T: **3.3.45 public-key**

页 6: [91] 批注 [P91] POOL

Page: 6
D: (In a public key cryptosystem) that key of a user's key pair which is publicly known.

页 6: [92] 批注 [P92] POOL

Page: 6
T: **3.3.46 public-key certificate (PKC)**

页 6: [93] 批注 [P93] POOL

Page: 6
D: The public key of a user, together with some other information, rendered unforgeable by digital signature with the private key of the certification authority which issued it.

页 6: [94] 批注 [P94] POOL

Page: 6
T: **3.3.47 public key infrastructure (PKI)**

页 6: [95] 批注 [P95] POOL

Page: 6
D: The infrastructure able to support the management of public keys able to support authentication, encryption, integrity or non-repudiation services.

页 6: [96] 批注 [P96] POOL

Page: 6
T: **3.3.48 relying party**

页 6: [97] 批注 [P97] POOL

Page: 6
D: A user or agent that relies on the data in a certificate in making decisions.

页 6: [98] 批注 [P98] POOL

Page: 6
T: **3.3.49 role assignment certificate**

页 6: [99] 批注 [P99] POOL

Page: 6
D: A certificate that contains the role attribute, assigning one or more roles to the certificate subject/holder.

页 6: [100] 批注 [P100] POOL

Page: 6
T: **3.3.50 role specification certificate**

页 6: [101] 批注 [P101] POOL

Page: 6
D: A certificate that contains the assignment of privileges to a role.

页 6: [102] 批注 [P102] POOL

Page: 6
T: **3.3.51 sensitivity**

页 6: [103] 批注 [P103] POOL

Page: 6
D: Characteristic of a resource that implies its value or importance.

页 6: [104] 批注 [P104] POOL

Page: 6
T: **3.3.52 simple authentication**

页 6: [105] 批注 [P105] POOL

Page: 6
D: Authentication by means of simple password arrangements.

页 6: [106] 批注 [P106] POOL

Page: 6
T: **3.3.53 security policy**

页 6: [107] 批注 [P107] POOL

Page: 6
D: The set of rules laid down by the security authority governing the use and provision of security services and facilities.

页 6: [108] 批注 [P108] POOL

Page: 6
T: **3.3.54 self-issued AC**

页 6: [109] 批注 [P109] POOL

Page: 6
D: An attribute certificate where the issuer and the subject are the same Attribute Authority. An Attribute Authority might use a self-issued AC, for example, to publish policy information.

页 6: [110] 批注 [P110] POOL

Page: 6
T: **3.3.55 self-issued certificate**

页 6: [111] 批注 [P111] POOL

Page: 6
D: A public-key certificate where the issuer and the subject are the same CA. A CA might use self-issued certificates, for example, during a key rollover operation to provide trust from the old key to the new key

页 6: [112] 批注 [P112] POOL

Page: 6
T: **3.3.56 self-signed certificate**

页 6: [113] 批注 [P113] POOL

Page: 6
D: A special case of self-issued certificates where the private key used by the CA to sign the certificate corresponds to the public key that is certified within the certificate. A CA might use a self-signed certificate, for example, to advertise their public key or other information about their operations.

NOTE – Use of self-issued certificates and self-signed certificates issued by other than CA are outside the scope of this Recommendation | International Standard

页 6: [114] 批注 [P114] POOL

Page: 6
T: **3.3.57 source of authority (SOA)**

页 6: [115] 批注 [P115] POOL

Page: 6
D: An Attribute Authority that a privilege verifier for a particular resource trusts as the ultimate authority to assign a set of privileges.

页 6: [116] 批注 [P116] POOL

Page: 6
T: **3.3.58 strong authentication**

页 6: [117] 批注 [P117] POOL

Page: 6
D: Authentication by means of cryptographically derived credentials.

页 6: [118] 批注 [P118] POOL

Page: 6
T: **3.3.59 trust**

页 6: [119] 批注 [P119] POOL

Page: 6
D: Generally, an entity can be said to "trust" a second entity when it (the first entity) makes the assumption that the second entity will behave exactly as the first entity expects. This trust may apply only for some specific function. The key role of trust in this framework is to describe the relationship between an authenticating entity and an authority; an entity shall be certain that it can trust the authority to create only valid and reliable certificates

页 6: [120] 批注 [P120] POOL

Page: 6
T: **3.3.60 trust anchor**

页 6: [121] 批注 [P121] POOL

Page: 6
D: A trust anchor is a set of the following information in addition to the public key: algorithm identifier, public key parameters (if applicable), distinguished name of the holder of the associated private key (i.e., the subject CA) and optionally a validity period. The trust anchor may be provided in the form of a self-signed certificate. A trust anchor is trusted by a certificate using system and used for validating certificates in certification paths.

页 6: [122] 批注 [P122] POOL

Page: 6
A: AA Attribute Authority

页 6: [123] 批注 [P123] POOL

Page: 6
A: AARL Attribute Authority Revocation List

页 6: [124] 批注 [P124] POOL

Page: 6
A: AC Attribute Certificate

页 6: [125] 批注 [P125] POOL

Page: 6
A: ACRL Attribute Certificate Revocation List

页 6: [126] 批注 [P126] POOL

Page: 6
A: CA Certification Authority

页 6: [127] 批注 [P127] POOL

Page: 6
A: CARL Certification Authority Revocation List

页 7: [128] 批注 [P134] POOL

Page: 7
A: EARL End-entity Attribute certificate Revocation List

页 7: [129] 批注 [P135] POOL

Page: 7
A: EPRL End-entity Public-key certificate Revocation List

页 7: [130] 批注 [P136] POOL

Page: 7
A: iCRL Indirect Certificate Revocation List

页 7: [131] 批注 [P137] POOL

Page: 7
A: OCSP Online Certificate Status Protocol

页 7: [132] 批注 [P138] POOL

Page: 7
A: PKC Public-Key Certificate

页 7: [133] 批注 [P139] POOL

Page: 7
A: PKCS Public-Key Cryptosystem

页 7: [134] 批注 [P140] POOL

Page: 7
A: PKI Public-Key Infrastructure

页 7: [135] 批注 [P141] POOL

Page: 7
A: PMI Privilege Management Infrastructure