This electronic version (PDF) was scanned by the International Telecommunication Union (ITU) Library & Archives Service from an original paper document in the ITU Library & Archives collections.

La présente version électronique (PDF) a été numérisée par le Service de la bibliothèque et des archives de l'Union internationale des télécommunications (UIT) à partir d'un document papier original des collections de ce service.

Esta versión electrónica (PDF) ha sido escaneada por el Servicio de Biblioteca y Archivos de la Unión Internacional de Telecomunicaciones (UIT) a partir de un documento impreso original de las colecciones del Servicio de Biblioteca y Archivos de la UIT.

هذه النسخة الإلكترونية (PDF) نتاج تصوير بالمسح الضوئي أجراه قسم المكتبة والمحفوظات في الاتحاد الدولي للاتصالات (ITU) نقلاً من وثيقة ورقية أصلية ضمن الوثائق المتوفرة في قسم المكتبة والمحفوظات.

此电子版（PDF版本）由国际电信联盟（ITU）图书馆和档案室利用存于该处的纸质文件扫描提供。

Настоящий электронный вариант (PDF) был подготовлен в библиотечно-архивной службе Международного союза электросвязи путем сканирования исходного документа в бумажной форме из библиотечно-архивной службы МСЭ.

© International Telecommunication Union

国 际 电 信 联 盟

CCITT

国际电报电话咨询委员会

蓝 皮 书

卷 Ⅷ.8

数据通信网

号码簿

建议 X.500-X.521

第 九 次 全 体 会 议

1988年11月14—25日 墨尔本

1989年 日内瓦

国 际 电 信 联 盟

CCITT

国 际 电 报 电 话 咨 询 委 员 会

蓝 皮 书

卷 Ⅷ.8

数据通信网

号码簿

建议 X.500-X.521

第 九 次 全 体 会 议

1988年11月14—25日 墨尔本

1989年 日内瓦

ISBN 92-61-03735-6

ARCHIVES U.I.T. GENÈVE

© ITU

中国印刷

CCITT 图书目录
第九次全体会议（1988 年）

蓝 皮 书

卷 I

卷 I.1 — 全会会议记录和报告
研究组及研究课题一览表

卷 I.2 — 意见和决议
关于 CCITT 的组织和工作程序的建议（A 系列）

卷 I.3 — 术语和定义　缩略语和首字母缩写词　关于措词含义的建议（B 系列）和综合电信统计的建议（C 系列）

卷 I.4 — 蓝皮书索引

卷 II

卷 II.1 — 一般资费原则 — 国际电信业务的资费和帐务　D 系列建议（第 III 研究组）

卷 II.2 — 电话网和 ISDN — 运营、编号、选路和移动业务　建议 E. 100-E. 333（第 II 研究组）

卷 II.3 — 电话网和 ISDN — 服务质量、网络管理和话务工程　建议 E. 401-E. 880（第 II 研究组）

卷 II.4 — 电报业务和移动业务 — 运营和服务质量　建议 F. 1-F. 140（第 I 研究组）

卷 II.5 — 远程信息处理业务、数据传输业务和会议电信业务 — 运营和服务质量　建议F. 160-F. 353、F. 600、F. 601、F. 710-F. 730（第 I 研究组）

卷 II.6 — 报文处理和查号业务 — 运营和服务的限定　建议 F. 400-F. 422、F. 500（第 I 研究组）

卷 III

卷 III.1 — 国际电话连接和电路的一般特性　建议 G. 100-G. 181（第 XII 和 XV 研究组）

卷Ⅲ.2 — 国际模拟载波系统 建议 G.211-G.544（第XV研究组）

卷Ⅲ.3 — 传输媒质 — 特性 建议 G.601-G.654（第XV研究组）

卷Ⅲ.4 — 数字传输系统的概况；终端设备 建议 G.700-G.795（第XV和第XⅧ研究组）

卷Ⅲ.5 — 数字网、数字段和数字线路系统 建议 G.801-G.956（第XV和第XⅧ研究组）

卷Ⅲ.6 — 非话信号的线路传输 声音节目和电视信号的传输 H和J系列建议（第XV研究组）

卷Ⅲ.7 — 综合业务数字网（ISDN）— 一般结构和服务能力 建议 I.110-I.257（第XⅧ研究组）

卷Ⅲ.8 — 综合业务数字网（ISDN）— 全网概貌和功能、ISDN用户—网络接口 建议 I.310-I.470（第XⅧ研究组）

卷Ⅲ.9 — 综合业务数字网（ISDN）— 网间接口和维护原则 建议 I.500-I.605（第XⅧ研究组）

卷Ⅳ

卷Ⅳ.1 — 一般维护原则：国际传输系统和电话电路的维护 建议 M.10-M.782（第Ⅳ研究组）

卷Ⅳ.2 — 国际电报、相片传真和租用电路的维护 国际公用电话网的维护 海事卫星和数据传输系统的维护 建议 M.800-M.1375（第Ⅳ研究组）

卷Ⅳ.3 — 国际声音节目和电视传输电路的维护 N系列建议（第Ⅳ研究组）

卷Ⅳ.4 — 测量设备技术规程 O系列建议（第Ⅳ研究组）

卷Ⅴ — 电话传输质量 P系列建议（第XⅡ研究组）

卷Ⅵ

卷Ⅵ.1 — 电话交换和信令的一般建议 ISDN中服务的功能和信息流 增补 建议 Q.1-Q.118（乙）（第Ⅺ研究组）

卷Ⅵ.2 — 四号和五号信令系统技术规程 建议 Q.120-Q.180（第Ⅺ研究组）

卷Ⅵ.3 — 六号信令系统技术规程 建议 Q.251-Q.300（第Ⅺ研究组）

卷Ⅵ.4 — R1和R2信令系统技术规程 建议 Q.310-Q.490（第Ⅺ研究组）

卷Ⅵ.5 — 综合数字网和模拟—数字混合网中的数字本地、转接、组合交换机和国际交换机 增补 建议 Q.500-Q.554（第Ⅺ研究组）

卷Ⅵ.6 — 各信令系统之间的配合 建议 Q.601-Q.699（第Ⅺ研究组）

卷Ⅵ.7 — 七号信令系统技术规程 建议 Q.700-Q.716（第Ⅺ研究组）

卷Ⅵ.8 — 七号信令系统技术规程 建议 Q.721-Q.766（第Ⅺ研究组）

卷Ⅵ.9 — 七号信令系统技术规程 建议 Q.771-Q.795（第Ⅺ研究组）

卷Ⅵ.10 — 一号数字用户信令系统（DSS 1） 数据链路层 建议 Q.920-Q.921（第Ⅺ研究组）

卷Ⅵ.11 — 一号数字用户信令系统（DSS 1） 网络层、用户—网路管理 建议 Q.930-Q.940（第Ⅺ研究组）

卷Ⅵ.12 — 公用陆地移动网 与ISDN和PSTN的互通 建议Q.1000-Q.1032（第Ⅺ研究组）

卷Ⅵ.13 — 公用陆地移动网 移动应用部分和接口 建议Q.1051-Q.1063（第Ⅺ研究组）

卷Ⅵ.14 — 与卫星移动通信系统的互通 建议Q.1100-Q.1152（第Ⅺ研究组）

卷Ⅶ

卷Ⅶ.1 — 电报传输 R系列建议 电报业务终端设备 S系列建议 （第Ⅸ研究组）

卷Ⅶ.2 — 电报交换 U系列建议（第Ⅸ研究组）

卷Ⅶ.3 — 远程信息处理业务的终端设备和协议 建议T.0-T.63（第Ⅷ研究组）

卷Ⅶ.4 — 智能用户电报各建议中的一致性测试规程 建议T.64（第Ⅷ研究组）

卷Ⅶ.5 — 远程信息处理业务的终端设备和协议 建议T.65-T.101，T.150-T.390（第Ⅷ研究组）

卷Ⅶ.6 — 远程信息处理业务的终端设备和协议 建议T.400-T.418（第Ⅷ研究组）

卷Ⅶ.7 — 远程信息处理业务的终端设备和协议 建议T.431-T.564（第Ⅷ研究组）

卷Ⅷ

卷Ⅷ.1 — 电话网上的数据通信 V系列建议（第ⅩⅦ研究组）

卷Ⅷ.2 — 数据通信网：业务和设施，接口 建议X.1-X.32（第Ⅶ研究组）

卷Ⅷ.3 — 数据通信网：传输，信令和交换，网络概貌，维护和管理安排 建议X.40-X.181（第Ⅶ研究组）

卷Ⅷ.4 — 数据通信网：开放系统互连（OSI）— 模型和记法表示，服务限定 建议X.200-X.219（第Ⅶ研究组）

卷Ⅷ.5 — 数据通信网：开放系统互连（OSI）— 协议技术规程，一致性测试 建议X.220-X.290（第Ⅶ研究组）

卷Ⅷ.6 — 数据通信网：网间互通，移动数据传输系统，网间管理 建议X.300-X.370（第Ⅶ研究组）

卷Ⅷ.7 — 数据通信网：报文处理系统 建议X.400-X.420（第Ⅶ研究组）

卷Ⅷ.8 — 数据通信网：号码簿 建议X.500-X.521（第Ⅶ研究组）

卷Ⅸ — 干扰的防护 K系列建议（第Ⅴ研究组） 电缆及外线设备的其他部件的结构、安装和防护 L系列建议（第Ⅵ研究组）

卷Ⅹ

卷Ⅹ.1 — 功能规格和描述语言（SDL） 使用形式描述方法（FDT）的标准 建议Z.100和附件A、B、C和E 建议Z.110（第Ⅹ研究组）

卷Ⅹ.2 — 建议Z.100的附件D：SDL用户指南（第Ⅹ研究组）

卷 X.3 — 建议 Z.100 的附件 F.1：SDL 形式定义 介绍（第 X 研究组）
卷 X.4 — 建议 Z.100 的附件 F.2：SDL 形式定义 静态语义学（第 X 研究组）
卷 X.5 — 建议 Z.100 的附件 F.3：SDL 形式定义 动态语义学（第 X 研究组）
卷 X.6 — CCITT 高级语言（CHILL） 建议 Z.200（第 X 研究组）
卷 X.7 — 人机语言（MML） 建议 Z.301-Z.341（第 X 研究组）

蓝皮书　卷Ⅷ.8　目录

建议号　　页

X.500　号码簿概念、模型和服务概述 …… 3

X.501　号码簿模型 …… 19

X.509　号码簿验证框架 …… 48

X.511　号码簿抽象服务定义 …… 82

X.518　号码簿分布式操作规程 …… 116

X.519　号码簿协议描述 …… 174

X.520　号码簿属性类型集 …… 189

X.521　号码簿客体类别集 …… 212

卷　首　说　明

1　在 1989-1992 年度研究期中，委托给各研究组的研究课题已刊登在研究组的一号文稿中。

2　本卷中的“主管部门”一词是电信主管部门或经认可的私营业务代理（RPOA）的简称。

3　除特殊声明外，凡是 X 系列建议的附件和附录都应当作如下说明：
— 建议的附件是该建议整体的组成部分；
— 建议的附录不作为该建议的组成部分，而仅用来提供补充说明。

4　本卷收集的 X 系列建议是同 ISO/IEC 的合作共同制定的。这些建议对应的 ISO/IEC 标准之间的交叉索引列于下表。

CCITT 建议	ISO/IEC 标准
X.500	ISO 9594-1, Information processing systems - Open Systems Interconnection - The Directory - Part 1: Overview of concepts, models and service [a)]
X.501	ISO 9594-2, Information processing systems - Open Systems Interconnection - The Directory - Part 2: Models [a)]
X.509	ISO 9594-8, Information processing systems - Open Systems Interconnection - The Directory - Part 8: Authentication framework [a)]
X.511	ISO 9594-3, Information processing systems - Open Systems Interconnection - The Directory - Part 3: Abstract service definition [a)]
X.518	ISO 9594-4, Information processing systems - Open Systems Interconnection - The Directory - Part 4: Procedures for distributed operations [a)]
X.519	ISO 9594-5, Information processing systems - Open Systems Interconnection - The Directory - Part 5: Protocol specifications [a)]
X.520	ISO 9594-6, Information processing systems - Open Systems Interconnection - The Directory - Part 6: Selected attribute types [a)]
X.521	ISO 9594-7, Information processing systems - Open Systems Interconnection - The Directory - Part 7: Selected object classes [a)]

a) 目前处于国际标准草案（DIS）阶段。

卷 Ⅷ.8

建议 X.500 — X.521

数据通信网：号码簿

ARCHIVES
U.I.T.
GENÈVE

PAGE INTENTIONALLY LEFT BLANK

PAGE LAISSEE EN BLANC INTENTIONNELLEMENT

号码簿概念、模型和服务概述[1]

(1988 年订于墨尔本)

目　录

0　绪论 …… 4

1　应用范围 …… 4

2　参考文献 …… 4

3　定义 …… 5

3.1　OSI 参考模型定义 …… 5
3.2　基本号码簿定义 …… 5
3.3　号码簿模型定义 …… 5
3.4　分布式操作定义 …… 6

4　缩写词 …… 6

5　号码簿概述 …… 6

6　号码簿信息库 (*DIB*) …… 7

7　号码簿服务 …… 9

7.1　绪论 …… 9
7.2　服务的限制条件 …… 10
7.3　号码簿的查询 …… 10
7.4　号码簿的修改 …… 11
7.5　其他结果 …… 11

8　分布式号码簿 …… 11

8.1　功能模型 …… 12
8.2　组织模型 …… 12
8.3　模型的操作 …… 12

9　号码簿协议 …… 15

附件 *A* — 号码簿的应用 …… 15

A.1　号码簿的环境 …… 15
A.2　号码簿服务特性 …… 16
A.3　号码簿的应用模式 …… 16
A.4　一般应用 …… 18

---

① 建议 X.500 与 ISO 9594-1，“号码簿概念、模型与服务概述”是在紧密合作下制定的，而且在技术上是协调的。

0 绪论

0.1 本文和本系列其他文件的目的在于简化信息处理系统之间的互连，以便提供号码簿服务。全套系统，包括它们的号码簿信息，可视为一个整体，叫做号码簿。号码簿包含 的全部信息叫做号码簿信息库（DIB），其典型用途是简化客体之间以及同这些客体、或围绕这些客体（如应用实体、人员、终端和分布表）的通信联系。

0.2 号码簿在开放系统互连中扮演重要角色；开放系统的目的是除使用互连标准之外，设法只通过极少的技术协议就能实现下列信息处理系统之间的互连：

— 不同厂家；
— 不同管理部门；
— 不同复杂程度；以及
— 不同年代。

0.3 本建议介绍号码簿与 DIB 的概念，提供它的模型和概述它的服务和功能。其他建议根据这些模型定义号码簿所提供的抽象服务和描述如何获得或扩展这种服务的有关协议。

1 应用范围

1.1 号码簿提供 OSI 应用、OSI 管理进程、其他 OSI 层实体以及通信服务所要求的号码簿功能。这些功能中，“对用户友好”的命名法可采用便于人类用户称呼的名称来指出某些客体（虽然不是所有的客体都需要取对用户友好的名称）；“名称对地址的映射”允许客体与其地址之间保持动态的对应关系。这种功能使 OSI 网络能够“自动调整”，即增加、删除和修改客体地址都不影响 OSI 网络操作。

1.2 号码簿的目的不是构成一个通用数据库系统，虽然号码簿本身可以建立在这样的系统上。举典型的通信号码簿为例，假设查询的频率同更新频率相比要高得多。予计更新频率受人和组织的动态影响的可能性要比受网络动态影响大。没有必要在全球范围内进行快速更新，因为同一个信息的新、旧版本同时存在的暂时情况是允许的。

1.3 号码簿的其中一个特点是：除了由于存取权的差别或只做局部更新（即不扩散更新）的原因外，查询结果与查询者的身份或地址无关。上述特性使号码簿不适合于某些通信应用，例如某些路由选择类型。

2 参考文献

建议 X.200 — 开放系统互连基本参考模型
建议 X.208 — 开放系统互连抽象语法符号第 1 集（ASN.1）的规范
建议 X.501 — 号码簿模型
建议 X.509 — 号码簿验证框架
建议 X.511 — 号码簿抽象服务定义
建议 X.518 — 号码簿分布式操作规程
建议 X.519 — 号码簿协议描述
建议 X.520 — 号码簿属性类型集
建议 X.521 — 号码簿客体类别集
建议 X.219 — 远程操作模型、符号与服务定义

3. 定义

本节的定义使用 § 4 中的缩写词。

3.1 *OSI* 参考模型定义

本建议根据建议 X.200 所述的概念编写，并使用下列术语：

a) *application-entity* 应用实体；

b) *Application Layer* 应用层；

c) *application process* 应用进程；

d) *application protocol data unit* 应用协议数据单元；

e) *application service element* 应用服务元素。

3.2 基本号码簿定义

a) *The Directory* 号码簿：一群相互合作的开放系统，其目的是提供号码簿服务；

b) *Directory Information Base* 号码簿信息库（*DIB*）：受号码簿管理的一套信息；

c) (*Directory*) *user* （号码簿）用户：号码簿的最终用户，即访问号码簿的实体或个人。

3.3 号码簿模型定义

本建议使用 X.501 建议定义的术语如下：

a) *Administration Directory Management Domain* 主管部门号码簿管理域；

b) *alias* 别名；

c) *attribute* 属性；

d) *attribute type* 属性类型；

e) *attribute value* 属性值；

f) *Directory Information Tree* (*DIT*) 号码簿信息树；

g) *Directory Management Domain* (*DMD*) 号码簿管理域；

h) *Directory System Agent* (*DSA*) 号码簿系统代理；

i) *Directory User Agent* (*DUA*) 号码簿用户代理；

j) *distinguished name* 可识别名；

k) *entry* 登录项；

l) *name* 名称；

m) *object* 客体（感兴趣的）；

n) *Private Directory Management Domain* 私营号码簿管理域；

o) *relative distinguished name* 相对可识别名；

p) *root* 根；

q) *schema* 体制；

r) *subordinate object* 下级客体；

s) *superior entry* 上级登录项；

t) *superior object* 上级客体；

u) *tree* 树。

3.4 分布式操作定义

本建议使用建议 X.518 定义的术语如下：

a) *chaining* 链锁转移；

b) *multicasting* 多点传播

c) *referral* 转移参考

4 缩写词

ADDMD	主管部门号码簿管理域
DAP	号码簿访问协议
DIB	号码簿信息库
DIT	号码簿信息树
DMD	号码簿管理域
DSA	号码簿系统代理
DSP	号码簿系统协议
DUA	号码簿用户代理
OSI	开放系统互连
PRDMD	私营号码簿管理域
RDN	相对可识别名

5 号码簿概述

5.1 号码簿是由一组开放系统构成的，这些开放系统以合作的方式保持有关实际世界中的一组客体的信息逻辑数据库。号码簿用户，包括人员和计算机程序，经过允许可以读取或修改信息或其中的一部分。每个用户在访问号码簿时由号码簿用户代理（DUA）代表出面，DUA 是一个应用程序。图 1/X.500 说明上述概念。

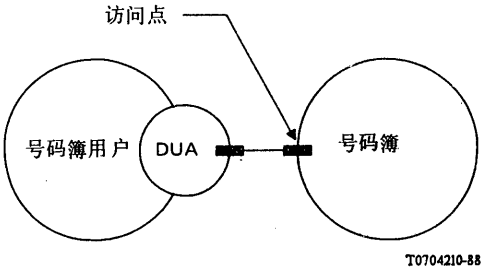


图 1/X.500

对号码簿的访问

注 — 本系列建议在说明号码簿时采用单数的、在统一的名称空间反映创造由许多系统构成和为许多应用提供服务的一个逻辑号码簿的意图。这些系统之间是否要进行合作则取决于它们所支持的应用的需要。处理无数不交叉客体的一些应用可能没有这个需要。一旦有这个需要，单一名称空间有利于上述系统之间进行合作。

5.2 存放在号码簿里的信息总称为号码簿信息库（DIB）。DIB 的结构在本建议的第 6 条款中描述。

5.3 号码簿向用户提供定义明确的一套访问功能，即号码簿的抽象服务。这种服务（在本建议 § 7 中描述）提供简单的修改功能和检索功能。这种服务可以在本地 DUA 功能上建立并提供端点用户所需要的功能。

5.4 号码簿有可能沿着功能线和组织线进行分布，而且分布的范围可能很广。§ 8 概述号码簿的相应模型。建立号码簿的各种模型的目的是为了使各个组成部分之间相互合作以形成有机的整体而提供一个框架。

5.5 号码簿的提供与使用，要求用户（实质上是各 DUA）和号码簿的各种功能的组成部分之间进行合作。在许多场合，这要求不同的开放系统中的各个应用进程进行合作，从而要求采用如 § 9 中所述的标准化应用协议来指导这种合作。

5.6 号码簿的设计要求考虑到广泛的可能性，以支持多种应用。号码簿所支持的应用的性质将确定哪些客体应列入号码簿，哪些用户需要访问信息，和他们将采用哪一种访问方式。应用的内容可以非常具体，例如为电子邮件提供分布表；也可以是一般的，例如“私人之间的通信号码簿”。号码簿为利用应用之间的共同性提供机会：

— 一个客体有可能同一个以上应用相关联；甚至同一个客体的同一个信息也有可能同多个应用相关联。

为了支持上述情况，定义了许多客体类别和属性类型，它们将得到广泛的应用。这些定义在建议 X.520 和 X.521 中给出。

— 在一系列应用中，号码簿的某些使用方式将有共同之处，这方面的论述在附件 A 中给出。

6 号码簿信息库（DIB）

注 — DIB 及其结构在建议 X.501 中定义。

6.1 DIB 是由客体的有关信息构成的。它是由（号码簿）登录项组成的，每一个登录项是由一个客体的有关信息群构成的。每一个登录项由一些属性组成；每一个属性具有一个类型和一个或多个值。在一个特定登录项中，属性类型与登录项所描述的客体类别有关。

6.2 DIB 的登录项是按树形结构排列的，即号码簿信息树（DIT），其中的极点代表登录项。靠近树顶的登录项（即离根部不远的地方）往往代表客体，例如国家或单位，而靠近树底的登录项则代表人或应用进程。

注 — 本建议所定义的服务只在树形结构 DIT 上操作，本建议不排除将来（需要时）会出现新的结构。

6.3 每一个登录项具有一个可识别名，该可识别名是唯一能够毫不含糊地识别登录项的名称。可识别名的这些特性是从信息的树形结构中生成的。登录项的可识别名是由它的上级登录项的可识别名以及从该登录项中特别指定的属性值（可识别值）构成的。

6.4 在树叶上有一些登录项是别名登录项，其余的项目均为客体登录项。别名登录项指向客体登录项，并为对应客体提供另选名的基础。

6.5 号码簿实施一套规则，保证经过多次修改之后 DIB 仍然保持完好。这些称为号码簿体制的规则能够避免登录项中包含与客体类别相互矛盾的错误属性类型，或者包含不符合属性类型的错误属性值，或者登录项中包含一些错误类别的子登录项。

6.6　图 2/X.500 示出了 DIT 概念和它的组成部分。

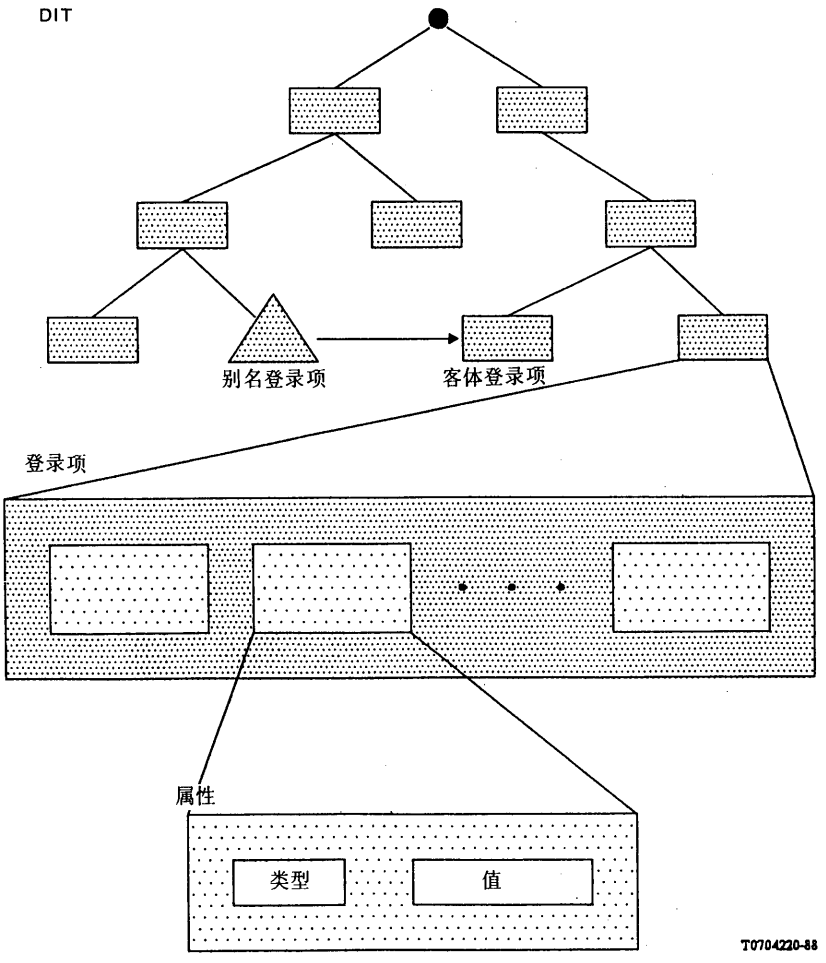


图 2/X.500

DIT 与登录项的结构

6.7　图 3/X.500 给出一个 DIT 的假想实例。这个树提供用来识别不同客体的一些属性类型的实例。例如，名称：

{C=GB，L=Winslow，D：绘图服务，CN：激光打印机}

表示应用实体“激光打印机”，在它的可识别名中有一个表明所在地的地理属性。居民 John Jones，他的名字叫做 GB，

{C=GB，L=Winslow，CN=John Jones}

在他的可识别名中也有一个相同地理属性。

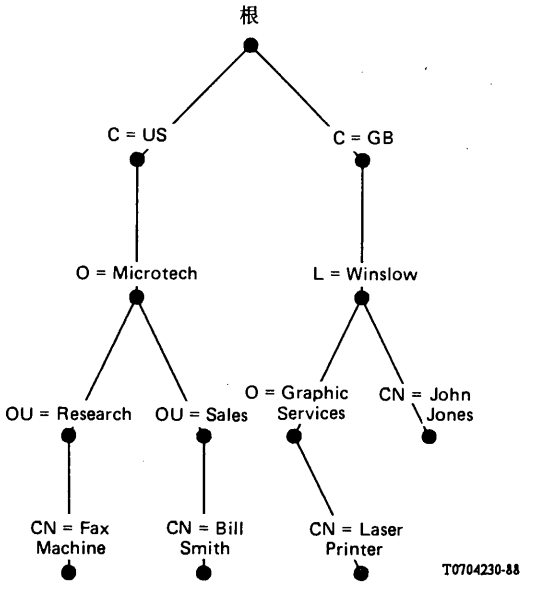


图 3/X.500

一个假想的号码簿信息树

6.8 DIT 的扩展与形状、号码簿体制的定义以及补充登录项时对可识别名的选择均由各个管理部门负责执行；管理部门的上下级关系反映在树的形状中。管理部门应当精心管理出现在名称中的属性类型和属性值，确保他们所管辖的所有登录项都有明确的可识别名。责任是从上级到下级管理部门，自上而下地沿着树传下去，并遵照号码簿体制进行控制。

7 号码簿服务

注 — 号码簿的抽象服务在建议 X.511 中定义。

7.1 绪论

7.1.1 本节概述号码簿向用户（由 DUA 代表）提供的服务。所有的服务均由号码簿在响应 DUA 请求时提供。向号码簿提出的请求有：查询（在 § 7.3 中说明）和修改（在 § 7.4 中说明）。此外，服务请求可带附加条件，如 § 7.2 中所述。号码簿必须对提出的每一个请求提供结果报告。正常结果的形式取决于具体的请求，并可从请求的说明中看得出来。大多数不正常结果对一些请求是一样的。几种可能性在 § 7.5 中给出。

7.1.2 最终的号码簿服务的许多方面不在本系列建议描述的标准中提供。因此，在有标准功能之前，需要提供一些相应的功能做为局部功能。这些功能包括下列各项：

— 增加和删除任意登录项，从而允许建立一个分布式号码簿；
— 访问控制的管理（即发放或收回某用户访问某一段信息的许可证）；
— 号码簿体制的管理；
— 智能信息的管理；
— 部分 DIB 的复制。

注 — 上述列表并不全面。

7.1.3 号码簿应保证，无论是由于号码簿服务请求的结果还是通过其他（本地）手段改变DIB，最终的DIB仍然服从号码簿体制的准则。

7.1.4 一个用户和号码簿，在某一段时间内，在号码簿的访问点上彼此耦合在一起。在耦合的时间内，用户和号码簿可以选择是否要彼此查明对方的身份。

7.2 服务的限制条件

7.2.1 服务控制

各种服务请求可应用一些控制，主要是为了用户能够限制号码簿，使它不超过资源利用极限。控制的范围包括：时间的长短、结果的大小、搜索范围、交互方式以及请求的优先级。

7.2.2 保密参数

每一个请求可以携带保护号码簿信息的保密机制信息。保密机制信息包括用户的各种保护请求、请求的数字签字以及协助合法用户鉴别签字所使用的信息。

7.2.3 过滤器

有许多请求，如其结果包括来自或有关若干登录项的信息，可以携带一个过滤器。过滤器提出一个或多个登录项必须满足的条件，以便获得响应并成为结果的一部分。这样就可以从得到响应的登录项中只留下相关的部分。

7.3 号码簿的查询

7.3.1 读

读请求是针对某一个登录项提出的，并从响应中获得该登录项的一些或全部属性的值。在仅仅希望获得响应中的某些属性的场合，DUA可提供感兴趣的属性类型。

7.3.2 比较

比较请求是针对某一个登录项的某一个属性提出的，并使号码簿检查给出的值 是否与该属性值匹配。

注 — 例如，这个功能可用于口令检查；该口令保存在号码簿里，可能不允许读取但可进行比较。

7.3.3 名单

名单请求使号码簿回送在DIT中被指名的某一个登录项的直接下级的名单。

7.3.4 搜索

搜索请求能使号码簿从存放在DIT某一处的并满足某一个过滤器的所有登录项中回送信息。从每一个登录项中回送的信息包含该登录项的某一些或全部属性，这与读的情况类似。

7.3.5 放弃

在执行查询请求的过程中提出的放弃请求，导致向号码簿报告：请求的发起者对该请求的执行已不再

感兴趣。这时号码簿可以停止处理该请求，也可以放弃已经得到的结果。

7.4 号码簿的修改

7.4.1 增加登录项

增加登录项请求导致 DIT 增加一个新叶登录项（如一个客体登录项或一个别名登录项）。

注 — 在当前的形式下，本服务的目的是为了增加将仍然保持为叶子的登录项，例如人员的登录项或应用实体的登录项，而不是重复使用本服务来增加整个子树。预计本服务将会扩展以适应更为普遍的情况。

7.4.2 删除登录项

删除登录项请求导致从 DIT 上删除一个叶登录项。

注 — 与增加登录项请求类似，目前本服务的目的是为了在“真叶”登录项上操作，而且将加以扩展以适应一般情况。

7.4.3 修改登录项

修改登录项请求导致号码簿对某一个登录项完成一系列修改。要么全部修改，要么不修改任何部分，而且 DIB 永远保持符合于体制的状态。允许修改的内容包括增加、删除、更换属性或属性值。

7.4.4 修改相对可识别名

修改相对可识别名（RDN）请求导致 DIT 中的叶登录项（客体登录项或别名登录项均可）的相对可识别名，根据所指出的可识别属性值而发生变化。

7.5 其他结果

7.5.1 差错

任何服务都可能失败，例如由于用户提供的参数有问题，在这种情况下将报告差错。在可能时应回送差错信息，以便协助排除故障。但是，在一般情况下，只报告号码簿碰到的第一个差错。除上述由于用户提供的参数（尤其是无效登录项名或属性类型）造成的故障之外，违反保密政策、体制准则和服务控制也可能引起差错。

7.5.2 转移参考

服务失败的原因可能是由于连接 DUA 的某一个服务访问点不太适合于用来执行请求，例如由于请求所涉及的信息（在逻辑上）离访问点很远。在此情况下，号码簿可以回送一个转移参考，指出 DUA 提出请求时可以使用的另一个访问点。

注 — 号码簿和 DUA 可以分别选择是否要使用转移参考或是否要使用连锁请求（见 § 8.3.3.2）。DUA 可借助于服务控制表示自己的选择。号码簿应做出使用哪 一种方法的最后决定。

8 分布式号码簿

注 — 号码簿的模型在建议 X.501 中定义，分布式号码簿的操作规程在建议 X.518 中描述。

8.1 功能模型

号码簿的功能模型示于图 4/X.500。

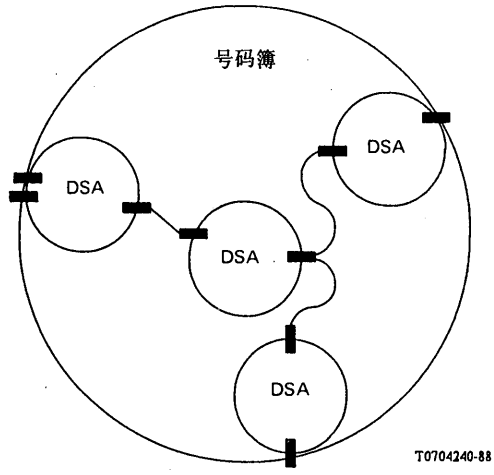


图 4/X.500
号码簿的功能模型

号码簿系统代理（DSA）是一个 OSI 应用进程，它是号码簿的一部分，其作用是向各 DUA 和/或其各 DSA 提供访问 DIB 的功能。DSA 可以使用存放在它的本地数据库的信息，或与其他各 DSA 互相配合执行请求。另一种办法是：DSA 可以将请求者指向另一个可协助完成请求的 DSA。本地数据库完全取决于实现方案。

8.2 组织模型

8.2.1 由一个组织管理的一个或多个 DSA 和零个或多个 DUA 可构成一个号码簿管理域（DMD）。有关组织在管理 DMD 内功能部件之间的通信时，可以使用也可以不使用本系列建议。

8.2.2 其他建议将描述各 DSA 特性的某些方面。为此，在一个 DMD 内部的一组 DSA 可根据 DMD 的管理组织的选择体现单个 DSA 的特性。

8.2.3 一个 DMD 可根据它是否在一个公众电信组织下经营，分为主管部门 DMD（ADDMD）或私营 DMD（PRDMD）。

注 — 必须明确，CCITT 成员对专用号码簿系统的支持受本国条例框架的约束。因此，这里描述的技术可能性不一定由提供号码簿服务的主管部门提供。私营 DMD 的内部操作和配置方案不属于本系列 CCITT 建议的讨论范围。

8.3 模型的操作

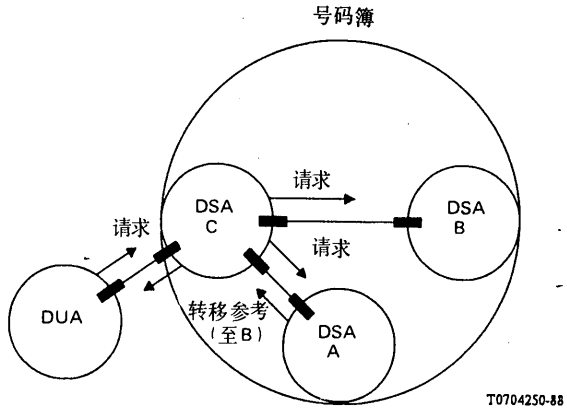
8.3.1 DUA 同号码簿之间的联系是通过 DUA 与一个或多个 DSA 之间的对话实现的。DUA 不需要固定于某一个 DSA，它在提出请求时可直接与各个 DSA 联系。由于某些行政上的原因，有时不可能与需要执行请求的 DSA 直接联系，例如回送某种号码簿信息。DUA 也有可能通过一个单一的 DSA 对号码簿进行访问。为此，各 DSA 需要彼此互相配合。

8.3.2 DSA 所关心的是执行 DUA 的请求和在所需要的信息没有时获得信息。为了获得信息，DSA 可以

DUA 的名誉与其他 DSA 进行联系。

8.3.3 建议讨论了许多有关处理请求的实例，如图 5-7/X.500 所示，并在下面进行了描述。

8.3.3.1 在图 5a/X.500 中，DSA C 由 DSA A 收到一个转移参考，并负责将请求运送给 DSA B（在 DSA A 提供的转移参考中指名），或将转移参考送回原 DUA。



注 - 假如DSA C 将转移参考回送给DUA，"请求（至B）"不会出现。同样地，假如DSA C将请求运送给DSA B，它将不对DUA回送转移参考。

图 5a/X.500

转 移 参 考

在图 5b/X.500 中，DUA 由 DSA C 收到转移参考，并负责将请求直接送到 DSA A（在 DSA C 提供的转移参考中指名）。

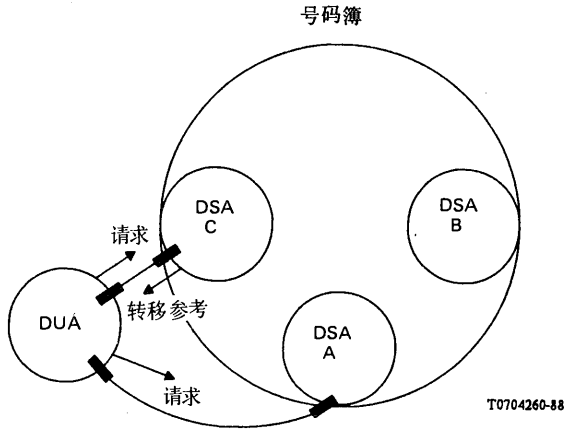


图 5b/X.500

转 移 参 考

8.3.3.2 图 6/X.500 示出 DSA 的连锁作用，通过这个作用可把请求经若干 DSA 往下传递直到回送响应为止。

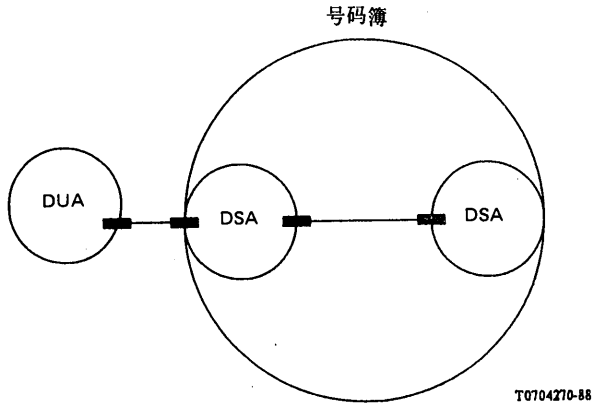


图 6/X.500

连 锁 方 式

8.3.3.3 图 7/X.500 示出多点传播方式。与 DUA 相联的 DSA，为了完成某个请求，把请求传给两个或多个 DSA；送到每一个 DSA 的请求是一样的。

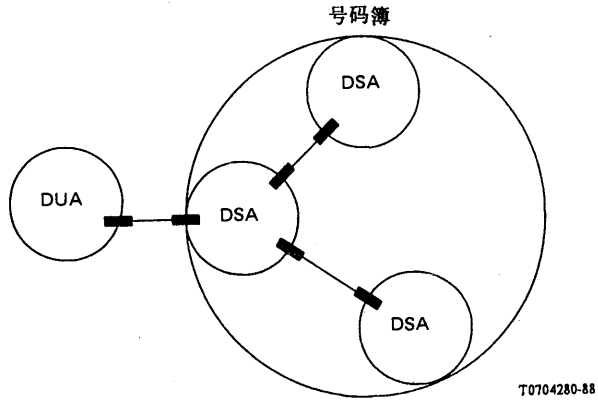


图 7/X.500

多点传播方式

8.3.4 上述方式各有优点。例如，图 5/X.500 的方式可用在需要为本地 DSA 减轻负载的场合。在其他场

合，为了满足发起者的需要可能要求采用更为复杂的功能间相互配合的混合方式，如图 8/X.500 所示。

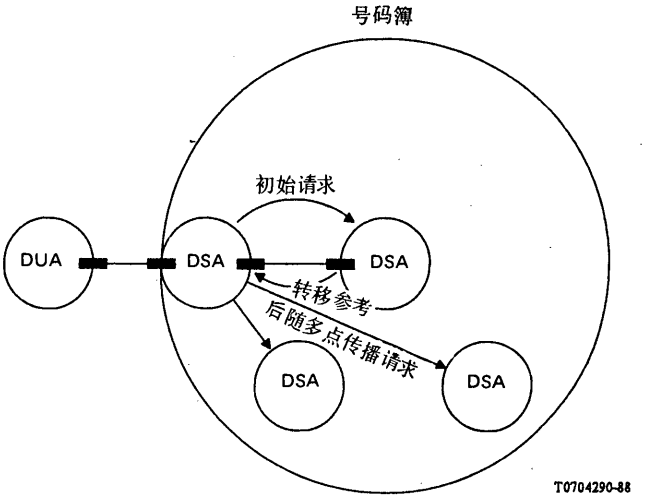


图 8/X.500

混 合 方 式

9 号码簿协议

注 — 允许 DUA 和 DSA 在不同的开放系统中进行合作的 OSI 应用层协议在建议 X.519 中描述。

9.1 有两个号码簿协议：

— 号码簿访问协议 Directory Access Protocol (DAP)：定义 DUA 与 DSA 之间的请求与结果的交换。
— 号码簿系统协议 Directory System Protocol (DSP)：定义两个 DSA 之间的请求与结果的交换。

9.2 每一个协议是由应用上下文定义的，每一个协议包含一套协议元素。例如，DAP 包含与查询和修改号码簿有关的协议元素。

9.3 每一个应用上下文由应用服务元素构成。这些应用服务元素是利用建议 X.219 的远程操作服务 (ROS) 定义的，目的是为了支持这些服务之间的相互合作和建立这种合作的结构。因此，利用 ROS 符号法把 DAP 和 DSP 定义为一套远程操作和差错。

附 件 A

(附于建议 X.500)

号码簿的应用

本附件不是本建议的一部分。

A.1 号码簿的环境

注 — 在本节中，“网络”一词泛指与任何电信服务有关的一套相互连接的系统和进程，而不是唯一与

OSI 网络层有关的网络。

号码簿存在于下列环境并在下列环境中提供服务：

a) 许多电信网的规模很大并且将不断变化：
   1) 各种客体不打招呼就进、出网，而且可以单独行动或集体行动；
   2) 由于客体之间的路径数会增加或减少，客体（尤其是网络节点）的连通性将有变化；
   3) 客体的各种特性，例如其地址、完好性和物理位置将随时发生变化；
b) 虽然总的变化率很高，但是任何客体的有用寿命并不短。通常，客体参与通信活动的次数要比它改变地址、完好性和物理位置等的次数高得多；
c) 参与当前电信服务的客体通常可通过号码或其他字符串来识别，选择号码或字符串是为了分配或处理上的方便，而不是为了方便人类。

A.2 号码簿服务特性

对号码簿功能的需要来自下列几项：

a) 希望把网络用户尽可能地与经常性的网络修改隔离开。为此可在用户与有关的客体之间插入一个“间接层”。这包括允许用户采用名称而不用地址来称呼客体。为此号码簿提供必要的映射服务；
b) 希望对用户提供更为“友好”的网络形象。例如，别名的使用、“黄页栏”（见 A.3.5）的提供等都能帮助用户减轻寻找和使用网络信息的负担。

号码簿允许用户获得有关网络的各种信息，并提供维护、分布和保护该信息的手段。

A.3 号码簿的应用模式

注 — 本子条款只关心号码簿的检索：假定号码簿修改服务仅仅是为了 DIB 在一段时间内保持适合于特定应用所需要的形式。

A.3.1 引言

在本系列标准里，号码簿服务是根据一个 DUA 所能提出的特定请求及其参数定义的。但是，当一个应用设计者考虑号码簿在该应用中对信息检索的要求时，他总是更多地考虑应用目标。因此，本条款论述了一些有关号码簿服务的高层次应用模式，这些模式很可能与许多应用有关。

A.3.2 查阅

号码簿的直接查阅可能是号码簿查询中最常用的类型。号码簿的直接查阅包括 DUA 提供客体的可识别名和属性类型。号码簿将回送与该属性类型相符的值。这是典型号码簿功能的概述，当请求的属性类型与某一个地址类型相符时就可获得这个功能。各种地址的属性类型都是标准化的，包括 OSI PSAP 地址、信息处理 O/R 地址以及电话和用户电报号码在内。

查阅得到读服务的支持，从而又进一步得到下列概述：

— 号码簿可以利用客体的可识别名之外的名称，例如别名进行查阅；

— 可以使用一个请求来查阅许多属性类型的值：在极端情况下，将回送登录项中的全部属性值。

A.3.3 对用户友好的命名法

给客体取名字时可以尽量做到使人类用户能够预测（或记忆）这些名字。具有这种特性的名字一般是由客体固有的属性而不是虚构的属性构成的。所有的应用都可以共用同一个客体名。

A.3.4 浏览

在许多面向人的号码簿使用中，用户（或DUA）可能不会直接叫出有关的客体的名字，不管这个名字对用户是否友好。但是该用户可能会“一目了然”。浏览功能允许人类用户围绕DIB寻找合适的登录项。

浏览是通过搜索服务和名单完成的，可能还配合读功能（虽然搜索服务本身包括读功能）来完成。

A.3.5 “黄页栏”

“黄页栏”类型的功能有好多种提供方式。最简单的方式是使用过滤器，采用有关特定属性的声明；该属性的值为“类”（如在建议X.520中定义的“商业类”属性类型）。这种方法除要求保证必要的属性必须存在以外，不要求在DIT中建立任何特殊的信息。但是，一般在搜索对象数量庞大的条件下进行搜索是很不经济的，因为过滤过程要求生成需要过滤的通用集合。

另一种方法是建立特殊的子树，该子树的命名结构是为了“黄页栏”类型搜索的需要而特别设计的。图A-1/X.500是一个“黄页栏”子树的例子，该子树只包含别名登录项。实际上，只要每一个储存在号码簿内的客体只拥有一个客体登录项，“黄页栏”子树内的登录项就有可能包含客体和别名登录项的混合体。

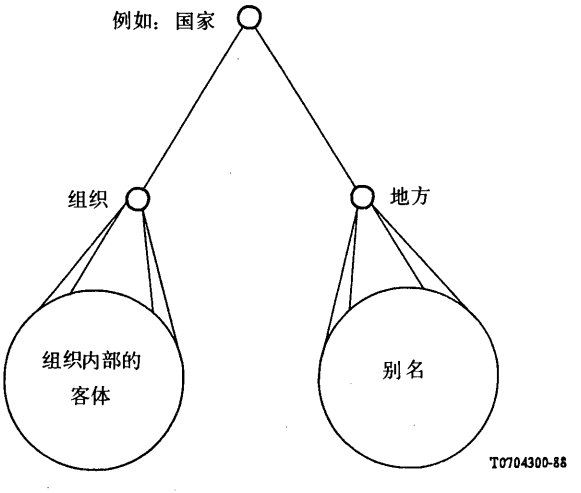


图 A-1/X.500

“黄页栏”的一个实例

A.3.6 群

群是一个集合，其成员因在一段时间内明显增加和减少而发生变化。该群是一个客体，同样的其成员也是客体。对号码簿可以请求：

— 指出某一客体是否为某一群的一个成员；

— 列出一个群的成员名单。

支持群的方法是群的登录项中提供多值“成员”属性（该属性类型在建议 X.520 中定义）。然后，通过比较和读就可以分别实现上述功能。

一个群的成员本身就可能是一个群，倘若这对应用有意义的话。但是，必要的循环证实服务和扩展服务须由 DUA 借助于提供给它的非循环证实服务和扩展服务建立。

A.3.7 验证法

许多应用要求相关客体在允许行动之前先证明自己的身份。号码簿为这种验证服务提供支援（做为独立实体，号码簿本身要求用户能够证明自己的身份来支持访问控制）。

较为直接的验证方法叫做“简单验证法”：号码簿在登录项中为准备向某一个服务证明自己的身份的用户提供“用户口令”属性。在服务的请求下，号码簿将确认或否认给出的某一个值为用户口令。这种方法避免用户需要对每一项服务使用不同的口令。倘若认为在使用简单验证法的本地环境里不宜交换口令，则号码簿可选择是否要提供保护口令的措施，避免单方面重报或滥用口令。

较为复杂的方法叫做“强验证法”：使用公用钥匙密码技术使号码簿成为存放公用密码钥匙，以防擅自动用的保险箱。建议 X.509 详细讨论用户如何从号码簿之中彼此获得公用钥匙，然后使用这些钥匙彼此互相验证的步骤。

A.4 一般应用

A.4.1 绪论

有许多一般应用可以设想为被号码簿以隐含的方式支持的：即不属于任何具体的通信服务的应用。建议中描述了两种这样的应用：人与人之间的通信号码簿和系统之间的通信号码簿（开放系统互连）。

注 — 在前面的条款里作为“访问模式”描述的验证法可以另外被设想为号码簿的一般性应用。

A.4.2 人与人之间的通信

本应用的目的是为人类或他们的代理提供有关怎样与其他人（或一群人）通信的资料。

下列客体类别是肯定要参与的：人员、组织角色和集团。有许多以比较间接的方式参与的其他类别，包括：国家、组织、单位。

除了在命名中使用的属性外，通常所关心的是寻址属性。某一个人的典型登录项通常包括和他联络时可使用的通信方式的相关地址，这些通信方式可从一个可扩展的一览表中选择，该表至少包括：电话、电子邮件、用户电报、ISDN、物理投递（例如邮政系统）、传真电报等。在某些情况下，例如电子邮件，该登录项还应包括一些附加信息，如用户设备所能处理的信息类型。倘若需要支持验证，则必须提供口令和/或凭证。

对各种客体类别使用的命名方案应对用户友好，例如建立适当的别名以提供另选名，或便于改名之后保持连续性等等。

在本应用中将阐述的访问模式是：查阅、对用户友好的命名法、浏览、“黄页栏”和群。在不同程度上也将使用验证功能。

A.4.3 *系统之间的通信（为 OSI）*

根据OSI参考模型需要两个号码簿功能：其中一个在应用层内操作——将应用名称映射到表示地址上；另一个在网络层内操作——将NSAP地址映射到SNPA地址上（SNPA＝子网连接点）。

注 — 本节其余部分只涉及应用层实例。

倘若不易用其他手段获得映射所需要的信息则可以通过号码簿的办法来完成上述功能。

用户是应用实体，感兴趣的客体类别也是应用实体，或是它的子类别。

主要的属性类型，除用于命名的那些属性类型之外，是表示地址。对号码簿功能本身认为不必要的其它属性类型能够支持验证或找出应用实体类型，以及所支持的应用上下文、抽象语法等的表格。也可能用上与验证有关的属性类型。

本应用中要阐述的主要访问模式是查阅。

建　　议 X.501

号码簿模型[①]

（1988年订于墨尔本）

目　　录

0　绪论…… 20

1　应用范围…… 21

2　参考文献…… 21

3　定义…… 21

4　缩写词…… 21

第1部分 — 号码簿模型 …… 22

5　号码簿模型…… 22

① 建议X.501与ISO 9594-2号码簿模型是在紧密合作下制定的，而且在技术上是协调的。

第 2 部分 — 信息模型 …… 24

6 号码簿信息库…… 24

7 号码簿登录项…… 26

8 名称…… 28

9 号码簿体制…… 31

第 3 部分 — 保密模型 …… 38

10 保密 …… 38

附件 *A* — 树的数学规则 …… 39

附件 *B* — 客体标识符的用法 …… 40

附件 *C* — ASN.1 中的信息框架 …… 41

附件 *D* — 定义索引表 …… 44

附件 *E* — 名称设计准则 …… 45

附件 *F* — 访问控制 …… 46

0 绪论

0.1 本文和本系列其他文件的目的在于简化信息处理系统之间的互连，以便提供号码簿服务。全套系统，包括它们的号码簿信息，可视为一个整体，叫做号码簿。号码簿包含的全部信息叫做号码簿信息库（DIB），其典型用途是简化客体之间以及同这些客体，或围绕这些客体（如应用实体、人员、终端和分布表）的通信联系。

0.2 号码簿在开放系统互连中扮演重要角色；它的目的是除使用互连标准之外，设法只通过使用极少的技术协议就能实现下列信息处理系统之间的互连：

— 不同厂家；
— 不同管理部门；
— 不同复杂程度；
— 不同年代。

0.3 本建议为其他建议框架提供不同的号码簿模型。这些模型包括总体（功能）模型、组织模型、保密模型和信息框架。信息框架描述号码簿如何组织它所掌握的信息。例如描述如何将有关客体的信息加以分类，为那些客体提供号码簿登录项，以及描述该信息如何为客体提供名称。

0.4 附件 A 概括地说明与树形结构有关的数学名词。

0.5 附件 B 概括地说明本系列建议使用 ASN.1 客体标识符的方法。

0.6 附件C提供ASN.1模块，包括与信息框架有关的所有定义。

0.7 附件D提供本建议定义的术语索引表。

0.8 附件E提供名称设计准则。

0.9 附件F提供访问控制准则。

1 应用范围

1.1 本建议提供的模型为描述号码簿的各个方面的其他建议提供概念性的和术语的框架。

1.2 功能模型与组织模型描述号码簿系统在功能上和管理上的分布方式。

1.3 保密模型定义提供号码簿保密功能（例如访问控制）的框架。

1.4 信息模型描述号码簿信息库（DIB）的逻辑结构。从这个角度看不出号码簿是分布式而不是集中式这一特征。本系列的其他建议将使用信息框架的概念。具体例子如下：

a) 号码簿提供的服务是引用信息框架的概念描述的（建议X.511）：这样提供的服务就比较地独立于DIB的物理分布；

b) 为了提供号码簿服务，建议X.518描述了号码簿的分布式操作；由于DIB实际上是高度分布式的，因此号码簿也应保持其逻辑信息结构。

2 参考文献

建议X.200 — 开放系统互连基本参考模型
建议X.500 — 号码簿概念、模型和服务概述
建议X.509 — 号码簿验证框架
建议X.511 — 号码簿抽象服务定义
建议X.518 — 号码簿分布式操作规程
建议X.519 — 号码簿协议描述
建议X.520 — 号码簿属性类型集
建议X.521 — 号码簿客体类别集

3 定义

名词术语的定义在各个适当章节的开始处给出。为便于参阅，附件D中给出了这些名词术语的索引。

4 缩写词

ADDMD 主管部门号码簿管理域
AVA 属性值判断语
DIB 号码簿信息库
DIT 号码簿信息树
DMD 号码簿管理域

DSA　　号码簿系统代理
DUA　　号码簿用户代理
PRDMD　私营号码簿管理域
RDN　　相对可识别名

第 1 部分 — 号码簿模型

5 号码簿模型

5.1 定义

a) *access point* 访问点：可获得抽象服务的地点。

b) *Administration Directory Management Domain* (*ADDMD*)　主管部门号码簿管理域：由主管部门管理的号码簿管理域（DMD）。

注 —“主管部门”是指公众电信主管部门或其他提供公众电信业务的组织；

c) *Administrative Authority*　行政权威：对所有储存在号码簿系统代理内的登录项拥有行政控制权的实体；

d) *Directory*　号码簿：贮存客体信息的贮藏室，并为其用户提供获得该信息的号码簿服务；

e) *Directory Management Domain* (*DMD*)　号码簿管理域：由一个组织管理的一个或多个 DSA 和零，或多个 DUA 的集合；

f) *Directory System Agent* (*DSA*)　号码簿系统代理：开放系统互连的一个应用进程，它是号码簿的一部分；

g) (*Directory*) *User*　（号码簿）用户：号码簿的终端用户，即访问号码簿的实体或人员；

h) *Directory User Agent* (*DUA*)　号码簿用户代理：开放系统互连的一个应用进程，它在访问号码簿过程中代表某一个用户；

注 — DUA 还能够提供一系列本地设施，协助用户提出查询和对响应进行解释；

i) *Private Directory Management Domain* (*PRDMD*)　私营号码簿管理域：除主管部门之外的组织管理的一个 DMD。

5.2 号码簿及其用户

5.2.1　号码簿用户（例如一个人或一个应用进程）通过访问号码簿获得号码簿服务。更确切地说，实际上是由一个号码簿用户代理（DUA）为某一个用户访问号码簿并与之互相配合来获得号码簿服务。号码簿提供一个或多个访问点并在该点上提供号码簿服务。图 1/X.501 绘出上述概念的示意图。

5.2.2　号码簿提供的服务在建议 X.511 中描述。

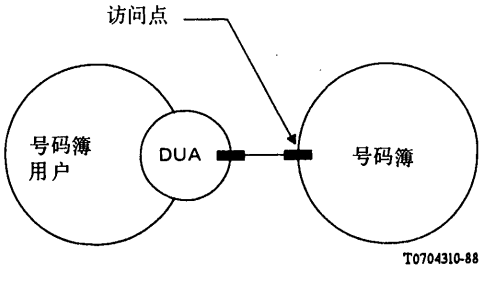


图 1/X.501

访问号码簿

5.2.3 号码簿是客体的登录项的一个贮藏室，该号码簿向用户提供的服务是获得这些登录项的各种访问方式。该登录项的综合总称为*号码簿信息库*（*DIB*）。DIB 的模型在本建议的第 2 部分给出。

5.2.4 DUA 被表现为一个应用进程。每一个 DUA 代表一个号码簿用户。

注 1 — 某些开放系统有可能为实际用户（应用进程、人员等）提供获得检索信息的集中式 DUA 功能。这功能对号码簿是透明的。

注 2 — 一些 DUA 功能和一个 DSA（见 § 5.3.1）有可能存在于同一个开放系统中，在实现上可以选择是否要让一个或多个 DUA 在开放系统互连的环境内成为可见的应用实体。

注 3 — DUA 有可能表现本系列建议之外的局部特性和结构。代表一个号码簿用户（例如，人）的 DUA 有可能提供一系列局部设施来协助它的用户提出查询和解释响应的内容。

5.3 功能模型

5.3.1 号码簿由称为*号码簿系统代理*（DSA）的一个或多个应用进程构成；每一个 DSA 提供零个、一个或多个访问点，如图 2/X.501 所示。由多个 DSA 构成的号码簿称为分布式号码簿。分布式号码簿的操作规程在建议 X.518 中给出。

注 — DSA 有可能表现本系列建议之外的局部特性和结构。例如，负责在 DIB 中保存一些或全部登录项的一个 DSA 通常使用数据库来完成这个任务，DSA 与该数据库的接口则属于局部决策。

5.3.2 在提供号码簿服务时需要互相配合的一对应用进程（例如一个 DUA 和一个 DSA，或两个 DSA）可置于不同的开放系统之中。上述应用进程互相配合时使用建议 X.519 中描述的开放系统互连号码簿协议。

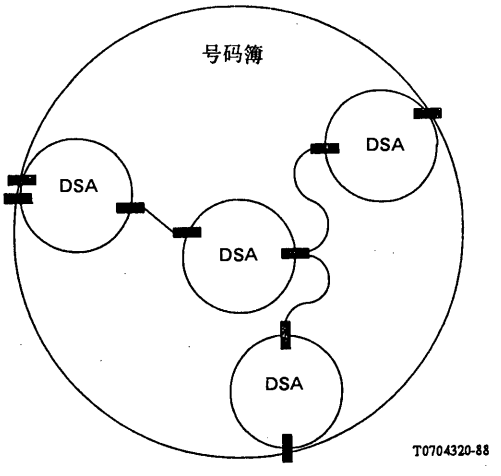


图 2/X.501

由多个 DSA 构成的号码簿

5.4 组织模型

5.4.1 由一个组织管理的一个或多个 DSA 和零个或多个 DUA 可构成一个*号码簿管理域*（*DMD*）。

注 — 负责管理 DMD 的组织可由一个主管部门（即一个公众电信主管部门或其他提供公众电信服务的组织）担任，在此情况下的 DMD 叫做主管部门 DMD（ADDMD）；或由一个私营 DMD（PRDMD）担任。必须指出，CCITT 成员为私营号码簿提供的支持不能超出本国条例的框架。因此，提供号码簿服务的主管部门有权决定是否提供前述技术可能性。本系列的建议不包括私营 DMD 的内部操作和安排。

5.4.2 DMD 对 DUA 的管理意味着 DMD 为该 DUA 提供服务所承担的当前责任，例如提供维护，或在某些场合指 DMD 的所有权等。

5.4.3 有关的组织可以决定选择使用或不使用本系列建议来控制完全存在于 DMD 内的 DUA 和 DSA 之间的交互作用。

5.4.4 每一个 DSA 由一个行政权威管理。这个管理实体负责管理该 DSA 所贮存的所有客体的登录项和别名登录项，并负责管理使用号码簿体制来指导如何产生和修改登录项（见 § 9）。命名权威的任务是对名称的结构和分配进行管理［见 § 8.1f)］，而行政权威的任务是实现号码簿体制内的命名结构。

第 2 部分 — 信息模型

6 号码簿信息库

6.1 定义

a) *alias entry* 别名登录项：是“别名”类别的一个登录项，它包含用于为某一个客体取别名的信息；

b) *Directory Information Base (DIB)* 号码簿信息库：是号码簿供用户访问的全套信息，它包含通过号码簿操作可以阅读或处理的全部信息单元；

c) *Directory Information Tree (DIT)* 号码簿信息树：DIB 被设想为一棵树；树的极点（除根之外）是该号码簿的登录项；

注 — 仅当涉及树形信息结构时才使用 DIT 而不使用 DIB。

d) *(Directory) entry* （号码簿）登录项：是 DIB 的一部分，它包含关于客体的信息；

e) *immediate superior* 直接上级（名词）：是相对于某一个登录项或客体（必须在所希望的上下文中明确）的上一级登录项或客体；

f) *immediately superior* 直接上级

（登录项）：是相对于某一登录项的一个登录项，它位于 DIT 的一个弧线初始极点，其最后的极点是某登录项的极点；

（客体）：是相对于某一客体的一个客体，该客体的登录项是第二客体的任意一个登录项（客体或别名）的直接上级；

g) *object (of interest)* 客体（感兴趣的）：是存在于某种“领域”的一种东西，该领域通常是指通信、信息处理或其中的某一部分，这个领域是可识别的（即可以命名），而且人们有兴趣在 DIB 内保存有关它的信息；

h) *object class* 客体类别： 是可识别的一族客体（或可设想的客体），它们具有某些共同的特征；

i) *object entry* 客体登录项：是贮存于 DIB 中的、有关某一客体主要信息的登录项，从而可认为它是客体在 DIB 中的代表；

j) *subclass* 子类别：是相对于超级类别而言、从超级类别中派生的客体类别，其成员具有超级类别的全部特征以及超级类别成员所没有的附加特征；

k） *subordinate/inferior* 下级/低级：是上级的反面；

l） *superclass* 超级类别：是相对于子类别的一个客体类别；超级类别派生出一个子类别；

m） *superior* 上级（用来形容登录项或客体）：是指直接上级，或直接上级的上级（循环出现）。

6.2 客体

6.2.1 号码簿的目的是保存和提供有关存在于某一“领域”的使人感兴趣的客体（客体）的信息。一个客体可以是存在于该可识别（可命名）领域里的任何东西。

注 1 — 所谓“领域”是泛指通信、信息处理或其中的某一部分。

注 2 — 号码簿所认识的客体不一定与存在于现实中的“实际”东西完全相符。举例来说，一个现实中的人在相关的号码簿里可视为两种不同的客体，例如一个商务人员或一个居民。本建议不对对照表定义，这个定义由号码簿提供者和用户在应用中确定。

6.2.2 号码簿所提供的全套信息统称为号码簿信息库（DIB）。通过号码簿操作可以阅读或处理的所有信息单元都应保存在 DIB 内。

6.2.3 客体类别是一个可识别的（或可设想的）客体群，它们具有某些共同的特性。每一个客体至少从属于某一个类别。一个客体类别可能是另一个客体类别的一个子类别：在此情况下可把该子类别的成员看成为该（超级）类别的成员。子类别的下面可以再派生子类别，依此类推。

6.3 号码簿登录项

6.3.1 DIB 是由多个号码簿登录项（登录项）构成的，每一个登录项包含说明某一客体的信息。

6.3.2 任何一个客体都有一个客体登录项，即关于该客体在 DIB 内的主要信息集合。客体登录项是客体的代表。

6.3.3 任何一个客体，除有一个客体登录项外还可以有一个或多个别名登录项为该客体提供其他名称（见§8.5）。

6.3.4 号码簿登录项的结构如图 3/X.501 所示，并在§7.2 中说明。

6.3.5 每一个登录项包含客体类别和该客体类别的超级类别的指示。对于客体登录项，这种指示是用来指明客体所属的类别。而对别名登录项，该指示是用来借助于某一个特殊的客体类别表示客体的“别名”（见§9.4.8.2），因为它实际上是一个别名登录项；另外它还可以用来指出该登录项所属的别名客体类别的子类别。

6.4 号码簿信息树（*DIT*）

6.4.1 为了满足大规模 DIB 的分布和管理要求以及保证客体能够有一个确切的名称（见§8），并保证找到

这些客体的登录项，登录项显然不宜采用平行的结构，而应采用客体等级式关系（例如，某人在某部门工作，该部门属于某一个组织，该组织设在某一个国家里），即将登录项布置成一棵树的形式，称为*号码簿信息树（DIT）*。

注 — 附件 A 给出树形结构概念和名词术语介绍。

6.4.2 DIT 的组成部分说明如下：

a） 极点表示登录项。客体登录项分为叶式和非叶式两种，但是别名登录项一定是叶式极点。树根不是登录项，但在适当场合［例如 b）和 c)］可把它看成零客体登录项［见下 d)］；

b） 弧线定义极点（即登录项）之间的关系。由极点 A 至极点 B 之间的弧线表示 A 点的登录项是 B 点的*直接上级登录项*（*简称为直接上级*），反过来 B 点的登录项是 A 点的直接下级登录项（简称为直接下级）。某一个登录项的*上级登录项*（简称上级）是该登录项的直接上级连同它的上级们（以此类推）。而某一个登录项的*下级登录项*（简称下级）是该登录项的直接下级连同它们的下级们（以此类推）；

c） 一个登录项所表示的客体与其下级的命名权威（见 § 8）有关或密切相关。

d） 根代表 DIB 命名权威的最高层。

6.4.3 客体之间的上、下级关系可从登录项之间的上、下级关系中寻找。仅当第一客体的客体登录项为第二客体的任何登录项的直接上级时，这一客体才称为另一客体的*直接上级客体*（*直接上级*）。*直接下级客体*、*直接下级*、*上级*和*下级*（对客体而言）等术语具有相似意义。

6.4.4 在客体之间，允许的上、下级关系受 DIT 结构定义的约束（见 § 9.2）。

7 号码簿登录项

7.1 定义

a） *attribute* *属性*：关于客体的具体类型的信息，在 DIB 中出现于描述该客体的登录项中；

b） *attribute type* *属性类型*：一个属性的一部分，它指明该属性，所提供的信息的类别；

c） *attribute Value* *属性值*：信息类别的一个具体实例，而信息类别是由其属性类型表示的；

d） *attribute value assertion* *属性值判断语*：一个命题，它可以为某一个登录项取真、假或不定值（或甚至只取可区分的值）。

注 — 在本文，表达式“string 1＝string 2”常用于描写属性值判断语的例子中。式中，“string 1”代表属性类型的缩写“名”，而“string 2”则代表某一适当值的文字表达形式。虽然举例中的属性类型常使用建议 X.520 定义的实在类型（例如“C”代表国家，CN 代表“普通名”），但对本建议而言并不要求这样做，因为号码簿通常并不知道使用属性类型的意义。

e） *distinguished value* *可识别值*：登录项的一个属性值，它应出现在登录项中的相对可识别名中。

7.2 总体结构

7.2.1 登录项是由一组*属性*构成的，如图 3/X.501 所示。

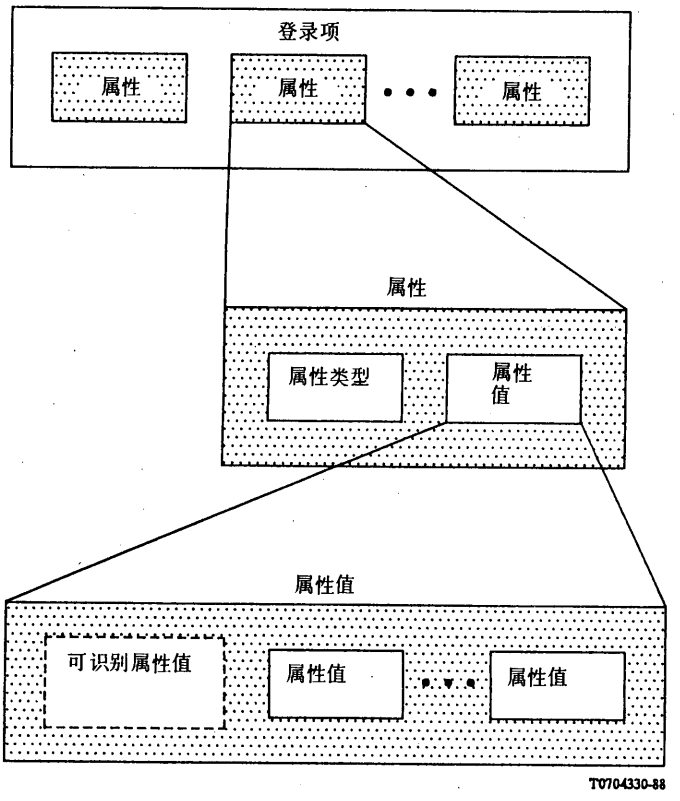


图 3/X.501
登录项结构示意图

7.2.2 每一个属性提供相应客体的信息，或说明相应客体的具体特征。

注 — 在登录项中可能出现的属性的例子包括命名信息，例如客体的人名和地址信息，或如电话号码。

7.2.3 属性是由一个属性类型和相应的属性值构成的；属性类型表明一个属性给出的信息类别；属性值表示出现在登录项中的类别的具体实例。

```
Attribute ::=
    SEQUENCE{
        type        Attribute Type
        values      SET OF AttributeValue
        -- at least one value is required --}
```

7.3 属性类型

7.3.1 一些属性类型将国际标准化；其他属性类型将由本国主管部门和私营组织定义。这意味着属性类型将由不同的权力机构负责分配，并保证每一类型都与其他类型不同。为此，定义类型时每一个属性类型都应有一个客体识别符（如§9.5所述）。

```
Attribute Type ::= OBJECT IDENTIFIER
```

7.3.2 在一个登录项中，所有属性都必须具有不同的属性类型。

7.3.3 号码簿掌握的和使用的属性类型有以下几种：

a) 客体类别（**ObjectClass**）：这种类型的属性出现在每一登录项中，并指出该客体所属的客体类别和超级类别。

b) 客体别名（**AliasedObjectName**）：这种类型的属性出现在每一个别名登录项中，并保存该别名登录项所述客体的可识别名（见§8.5）。

上述属性（一部分）的定义在§9.5.4中给出。

7.3.4 在登录项中必须出现或可能出现的属性的类型（§7.3.3所述除外）均受适用于指定客体类别的一些规则的约束。

7.4 属性值

7.4.1 定义属性值（见§9.5）时应当指明语法。即数据类型，而且这样的属性中的每一个值都必须符合上述语法。数据类型可取任意值：

```
AttributeValue ::= ANY
```

7.4.2 一个属性至多有一个值可指定为*可识别值*，此时属性值出现在登录项的相对可识别名中（见§8.3）。

7.4.3 *属性值判断语*（*AVA*）是一个命题，它把一个登录项值（或仅仅是可识别值）判为真、假或无定义。属性值判断语包含一个属性类型和一个属性值：

```
AttributeValueAssertion ::=
    SEQUENCE {AttributeType, AttributeValue}
```

而且：

a）当下列条件成立时属性值判断语为无定义：

i） 属性类型为未知；

ii） 类型的属性语法不包括等式匹配规则；

iii） 该值不符合属性语法的数据类型；

注 — ii）和iii）通常指出错误的AVA；但是i）可能只在局部发生（例如，有一个DSA未登记具体的属性类型）。

b）倘若该登录项包含该类型的一个属性，而且其中一值与该值相符（倘若判断语只关心可识别值，则相符的值一定是可识别值），则为真；

注 — 值的匹配是为了达到等同目的，它包括与属性语法有关的匹配规则。

c）在其他情况下则为假。

8 名称

8.1 定义

a） *alias*，*alias name* *别名*（*或别名名称*）：客体名，在DIT中使用一个或多个别名登录项时提供；

b） *dereferencing* *换名*：是用客体的可识别名取代客体的别名；

c） *distinguished name*（*客体的*）*可识别名*：客体名之一，它是由客体登录项和客体的每一个上级登录项的RDN序列构成的；

d） (*directory*) *name* （*号码簿*）*名*：从所有其他客体中区分某一特定客体的一个结构。名称必须明确（即只指出唯一的客体），但不一定要独一无二（即确切指出某一客体的名称不一定是唯一的）；

e） *purported name* *声称名*：一种结构，从语法上看是一个名称，但未经证实为一个有效名；

f） *naming authority* *命名权威*：负责分配名称的权威。每一客体，只要它的客体登录项位于DIT的

非叶式极点上，它就是一个命名权威，或与某一个命名权威密切相关；

g) *relative distinguished name* 相对可识别名（*RDN*）：与某一个登录项的可识别值有关的一组属性值判断语，每一个判断语皆为真。

8.2 名称的一般概念

8.2.1 （号码簿）名：用来从一组客体中识别某一特定客体的一种结构。每一个名称必须确切，即只指出唯一的客体。但是，每一个名称不一定是独一无二，即确切指出某一客体的名称不一定是唯一的。

8.2.2 从语法上讲，每一客体名是由一个有序的相对可识别名序列构成的（见§8.3）。

```
NAME ::=
    CHOICE { --only one possibility for now--
        RDNSequence}

RDNSequence ::= SEQUENCE OF RelativeDistinguishedName
DistinguishedName ::= RDNSequence
```

注 — 不按本规则而按其他方法构成的名称有可能成为将来的扩展名。

8.2.3 零序列是树根名。

8.2.4 一个客体名的每一个初始子序列同样也是一个客体名。按这个方法识别客体的序列是以树根开始，以被命名的客体而结束的，使得每一个序列中的客体成为位于它下面的客体的直接上级。

8.2.5 声称名是一种结构，从语法上看是一个名称，但并不表示一个有效名。

8.3 相对可识别名

8.3.1 每一个登录项都有一个独一无二的相对可识别名（*RDN*）。RDN 是与某一登录项的可识别值有关的属性值判断语集合，每一判断语皆为真。

```
RelativeDistinguishedName ::=
    SET OF AttributeValueAssertion
```

每一属性集合正好包含登录项中的每一个可识别值的一个判断语。

8.3.2 所有具有一个特定直接上级的登录项的 RDN 都彼此有别。与该登录项相关的命名权威的职责是保证上述要求付诸实施，为此必须为一个 RDN 指定适当的可识别属性值。

注 — 一个登录项往往包含一个可识别值（从而 RDN 包含一个 AVA）；但是，在某种情况下，为了区分，有可能使用附加值（从而使用 AVA）。

8.3.3 登录项的 RDN 是在建立登录项时选择的。任意属性类型的一个值实例可构成 RDN 的一部分，视所指出的客体类别的性质而定。RDN 的分配属于行政事务，为此在有关组织或主管部门之间可以（也可不）进行协商。本建议不提供协商机制和协商方式。必要时可用全部更换的办法来修改 RDN。

注 — 各 RDN 应当长期有效，号码簿用户可尽管贮存客体可识别名（例如在号码簿内部），而不必担心它们会过时。因此修改各 RDN 时应当十分谨慎。

8.4 可识别名

8.4.1 一给定客体的可识别名，定义为代表客体的登录项和它的全部上级登录项的 RDN 序列（向下递

减)。由于客体和客体登录项之间的一一对应关系，可认为一客体的可识别名也可用来识别客体登录项。

注 1 — 人们与之打交道的客体可识别名以对用户友好为宜。

注 2 — ISO 7498/3 定义原始名的概念。可识别名可用来做客体的原始名，因为：a） 可识别名是确切的，b）是唯一的，c）其内部结构（一个 RDN 序列）不需要（但当然可以）为号码簿的用户所理解。

注 3 — 由于客体可识别名只包含客体登录项及其上级，它永远不可能包含别名登录项。

8.4.2 定义根和别名登录项的“可识别名”是很方便的，而定义客体名和客体可识别名并不这样方便。根的可识别名定义为零序列。别名登录项的可识别名定义为别名登录项及其全部上级登录项（依次递减）的 RDN 序列。

8.4.3 图 4/X.501 给出 RDN 和可识别名的概念的实例。

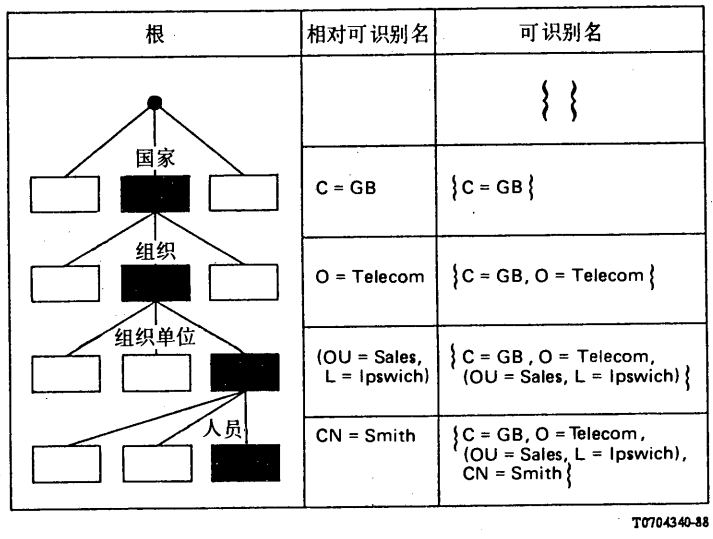


图 4/X.501

可识别名的构造

8.5 *别名*

8.5.1 客体的*别名*或*别名名称*是一个名称，它的 RDN 中至少有一个是别名登录项的 RDN。这些别名使客体登录项具有多个直接上级的效应。因此，别名提供了代替名的基础。

8.5.2 正如客体可识别名表示客体与某一个分级客体系统之间的主要关系，别名表示客体（在一般情况下）与不同的分级客体系统之间的另一种关系。

8.5.3 在 DIT 中，带有登录项的客体可有零或多个别名。因此，几个别名登录项可以指出同一个客体登录项。一个别名登录项可指出非叶登录项的一个客体登录项。只有客体登录项可以取别名。因此，不允许别名再取别名。

8.5.4 别名登录项不应当有下级，即别名登录项是一个叶登录项。

8.5.5 号码簿在别名登录项中使用客体别名属性来识别和寻找相应的客体登录项。

9 号码簿体制

9.1 定义

a) *Directory schema* 号码簿体制：用来描述 DIB 特性的 DIT 结构、客体类别定义、属性类型与语法的一套规则和约束条件。

b) *DIT structure rule DIT* 结构规则：号码簿体制的一个组成部分，它指出一个客体类别（下级）与另一个客体类别（上级）之间的依从关系，并在 DIT 中允许下级类别的登录项直接从属于上级类别之一。本规则还包括对允许出现在（下级）登录项的 RDN 中的属性类型的控制，并有可能提出附加条件。本体制可包含许多类似这样的规则。

9.2 概述

9.2.1 号码簿体制是关于 DIT 结构的一套定义和约束条件以及登录项的命名方法、可在登录项中保存的信息和描述该信息使用的属性等。

注 1 — 体制能使号码簿起下列作用：

— 避免建立错识的客体类别的下级登录项（例如将国家当做一个人的下级）；
— 避免把属性类型加到与客体类别不相符的一个登录项中（例如把一个序号加到一个人的登录项中）；
— 避免把一个与属性类型语法不相符的语法的属性值加上去（例如把一个可打印串加到一个比特串）；

注 2 — 本系列建议暂不提供号码簿体制的动态管理机制。

9.2.2 号码簿体制包含下列各项：

a) *DIT* 结构定义（规则）：定义登录项的可识别名以及它们在 DIT 中彼此间的关系的协调方法；

b) 客体类别定义：定义在某一给定类别的登录项中必须出现的必备属性和可出现的任选属性（见本建议 § 6.2.3）

c) 属性类型定义：定义客体的识别符，从这个客体的识别符可以识别一个属性，定义其语法和指出其是否可以取多值；

d) 属性语法定义：为每一个属性定义 ASN.1 数据类型和匹配规则。

图 5/X.501 给出体制定义之间的关系，DIT、号码簿登录项、属性以及属性值。

9.2.3 号码簿体制和 DIB 本身一样具有分布式的特征。每一个行政权威建立适用于它所管辖的那一部分 DIB 的号码簿体制。

注 — 本建议不支持体制信息在不同行政权威管理下的 DSA 的范围内的分布。这样的分布是由双方行

政权威协商进行的。

9.2.4 有关 DIT 结构、客体类别、属性类型以及属性语法的定义分别在§9.3 至§9.6 给出。

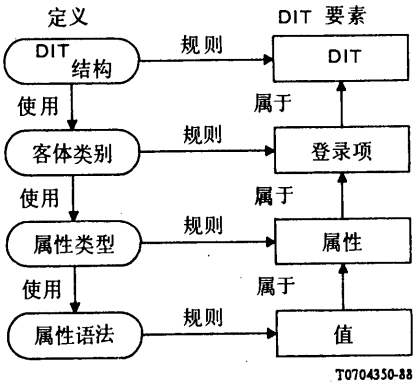


图 5/X.501

号码簿体制示意图

9.3 *DIT 结构定义*

9.3.1 DIT 结构规则定义登录项与其被允许的 RDN 之间的被允许的等级关系。DIT 结构的定义包含下列内容：

— 识别上、下级客体类别；

— 识别在下级登录项 RDN 中可能出现的属性类型；

—（任选的）附加信息。

9.3.2 号码簿允许一个登录项成为另一个登录项（它的直接上级）的直接下级的条件如下：仅当号码簿体制（见§9.2.3）中的一个 DIT 结构定义适用于包含该登录项的那一部分 DIB，而且：

— 该登录项属于下级客体类别；

— 该登录项的直接上级属于上级客体类别；

— 构成该登录项中的 RDN 的属性类型应当是被允许的属性类型；

— 附加信息元素附带的任何条件均得到满足。

注 1 — 本建议系列目前暂不提供描述 DIT 结构或代表 DIB 结构规则的方法。

注 2 — 倘若 DIT 结构规则允许属于某一个特定类别的上级或下级存在，则它也将以隐含的方式（除非代以明显的方式）允许由该类别派生的任何客体类别所属的上级或下级存在（见§9.4）。

9.3.3 号码簿对 DIT 中的每一个登录项规定明确的结构规则。任何违背现行结构规则修改 DIT 的尝试均不可能成功。

9.3.4 有关下级客体类别的 DIT 结构规则称为该客体类别的*名称耦合*。

9.3.5 为了使某部分 DIB 中的登录项能够代表一个客体类别，号码簿体制的适当部分至少应包含该客体类别的一个名称耦合。根据需要号码簿体制还可包含一些附加名称耦合。

注 — 在两个不同体制中出现的一个客体类别，可在这两个体制中分别使用不同的名称耦合。

9.4 客体类别定义

9.4.1 一个客体类别定义包含：

a) 任选项，为客体类别指定一个客体识别符；

b) 指出该客体是哪一个类别的子类别；

c) 除列出登录项的所有上级的必备属性类型之外，还应列出客体类别登录项应包含的必备属性类型。

d) 除列出登录项的所有上级的任选属性之外，还应列出客体类别登录项可能包含的任选属性类型。

注 — 设置无指定客体识别符的客体类别的目的是为了便于在本地给预定义的超级类别增加新的属性类型。给超级类别增加新的属性类型允许有多种可能性。例如，行政权威为了让用户能够给登录项增加任意属性，它可定义一个未登记的客体类别。行政权威可以限制特定客体类别的登录项使用的属性。例如，只允许使用本地属性表中的属性。除登记过的客体类别定义所要求的属性之外，行政权威还可以强制性地指定某一特定客体类别使用某些特定属性。

9.4.2 有一种特殊的客体类别，它所属的类别均为它的子类别。这个特殊客体类别的名字叫做“顶”（“Top”），并于§9.4.8.1中定义。

9.4.3 每一个登录项应包含客体类别 **ObjectClass** 类型的属性，来指出该登录项是属于哪一个客体类别和超级类别。这个属性的定义在§9.5.4给出。该属性是多值的。客体类别和它的每一个超级类别应有一个属性值，该值应包含一个客体标识符，但只要存在其他值，“顶”的值就不再需要了。

注 1 — 顶“Top”的定义反映每个登录项必须包含一个客体类别 **Object Class** 属性的要求。

注 2 — 由于认为一个客体类别从属于它所有的超级类别，因此超级类别链到顶的每一个成员都是通过客体类别属性中的一个值表示的（而且在这个链条中的任何一个值都可以用一个过滤器筛选）。

客体类别（**ObjectClass**）属性是由号码簿管理的，即用户不能自行修改。

9.4.4 号码簿对 DIB 中的每一个登录项均规定使用已定义的客体类别。任何违背登录项客体类别定义而修改登录项的企图均不可能成功。

注 — 尤其是号码簿将阻止下列行为：

a) 把不是客体类别定义中的属性类型加到该客体类别的一个登录项上；

b) 在缺少登录项的客体类别必备的一个或多个属性类型的情况下建立该登录项；

c) 删除登录项的客体类别的必备属性类型。

9.4.5 特殊的客体类别别名 **Alias** 在§9.4.8.2中定义。每个别名登录项应该有一个客体类别，它是本类别的子类别。

注 — 号码簿对别名登录项的换名保证别名登录项的客体类别 **ObjectClass** 属性值不经常暴露。建议从“**Alias**”中导出适当的别名客体类别，而不必指定客体识别符。

9.4.6 下列 **ASN.1** 宏可以（但不必须）用来定义一个客体类别。**SubclassOf** 的空乘积（empty production）仅在定义顶时允许存在：

```
OBJECT-CLASS MACRO ::=
BEGIN

TYPENOTATION ::= SubclassOf
                 MandatoryAttributes
                 OptionalAttributes
```

```
VALUENOTATION ::=
    value(VALUE OBJECT IDENTIFIER)

SubclassOf ::=
    "SUBCLASS OF" Subclasses |
    empty

Subclasses ::=    Subclass | subclass ","
                  Subclasses
Subclass ::=      value (OBJECT-CLASS)

MandatoryAttributes ::=
    "MUST CONTAIN {"Attributes"}" | empty

OptionalAttributes ::=
    "MAY CONTAIN {"Attributes"}" | empty

Attributes      ::= AttributeTerm | AttributeTerm "," Attributes

AttributeTerm   ::= Attribute | AttributeSet

Attribute       ::= value(ATTRIBUTE)

AttributeSet    ::= value(ATTRIBUTE-SET)

END
```

§ 9.4.1 所列定义的各部分与宏所引出的各种符号之间的对应关系如下：

a) 客体类别的客体标识符是宏在值分配中给出的值；

b) 超级类别（本客体类别是该超级类别的一个子类别）是用 **SubclassOf** 乘积，即列在“**SUBCLASS OF**”之后的各项表示的；

c) 必备属性是用 **MandatoryAttributes** 乘积给出的客体标识符，即列在“**MUST CONTAIN**”之后各项表示的；

d) 任选属性是用 **OptionalAttributes** 乘积给出的客体标识符，即列在“**MAY CONTAIN**”之后各项表示的。

注 1 — c）和 d）中的客体标识符是用来识别个别属性和属性集合的（见 § 9.4.7）。在两种情况下有效的表格是这些属性的集合或。倘若一个属性同时出现在必备集合和任选集合中，则应认为该属性是必备的。

注 2 — 宏用来定义建议 X・521 中的被选客体类别。

倘若上述描述于 b）、c）和 d）中的和宏给出的符号全部为空，则最终的符号（“**OBJECT-CLASS**”）可用来指出任何可能的客体类别。

9.4.7 一个属性集合是用一个客体标识符表示的一组属性。属性集合的定义如下：

a) 为某一集合指定一个客体标识符；

b) 列出属性和其他属性集合的客体标识符；该集合的成员形成一个集合。

下列 ASN.1 宏可以（但不必须）用来定义用于 **OBJECT-CLASS** 宏的一组属性：

```
ATTRIBUTE-SET-MACRO ::=
    BEGIN
    TYPE NOTATION ::= "CONTAINS" {"Attributes"}" | empty
    VALUE NOTATION ::= value(VALUE OBJECT IDENTIFIER)
```

```
Attributes ::=
    AttributeTerm | AttributeTerm "," Attributes
AttributeTerm  ::=  Attribute | AttributeSet
Attribute      ::=  value(ATTRIBUTE)
AttributeSet   ::=  value(ATTRIBUTE-SET)
END
```

一个属性集合的定义的各部分和宏给出的符号之间的对应关系如下：

a) 指定给属性集合的客体标识符是在宏的值分配中给出的值；

b) 包含属性集合的属性集是由属性的"集合或"和以 **Attributes** 乘积表示的属性集(即在"**CONTAINS**"之后的属性集)组成的。

倘若选择"**empty**"(空)符号，则最终的符号("**ATTRIBUTE-SET**)"可用来指定任何可能的属性集合。

9.4.8 前述客体类别在§9.4.8.1和§9.4.8.2定义。

注 — 这些都是偏定义：客体识别符实际上在建议X.521中被分配给这些客体类别，以便在本建议序列中提供这些客体识别符的单分配点。

9.4.8.1 客体类别"**Top**"(顶)定义如下：

```
Top ::=
    OBJECT-CLASS
        MUST CONTAIN {ObjectClass}
```

9.4.8.2 客体类别"**Alias**"(别名)定义如下：

```
Alias ::=
    OBJECT-CLASS
        SUBCLASS OF top
        MUST CONTAIN {aliasedObjectName}
```

注 1 — 客体类别"**Alias**"不为别名登录项的RDN指明适当的属性类型。行政权威可以通过指明类别"**Alias**"的子类别来指明别名登录项的RDN的有用属性类型(见建议X.521)。

注 2 —(别名)"**Alias**"的一个子类别的登录项是别名登录项。

9.5 属性类型定义

9.5.1 属性类型定义包含：

a) 给属性类型指定一个客体标识符；

b) 指出或定义属性类型的属性语法；

c) 指出此种类型是否可以只有一个或有多个值(循环)。

9.5.2 本号码簿保证被指出的属性语法使用于此种类型的每一个属性。本号码簿还保证倘若此种类型的属性被定义为只有一个值，则它将只有一个唯一的值。

9.5.3 下列ASN.1宏可以(但不必须)用来定义一个属性类型：

```
ATTRIBUTE MACRO ::=
BEGIN
TYPENOTATION      ::= AttributeSyntax Multivalued | empty
```

```
VALUENOTATION      ::= value (VALUE OBJECT IDENTIFIER)

AttributeSyntax    ::=
     "WITH ATTRIBUTE-SYNTAX" SyntaxChoice

Multivalued        ::= "SINGLE VALUE"
                       | "MULTIVALUE" | empty

SyntaxChoice       ::= value(ATTRIBUTE-SYNTAX)
                       Constraint | type MatchTypes

Constraint         ::= "("ConstraintAlternative")" | empty

ConstraintAlternative ::= StringConstraint | IntegerConstraint

StringConstraint   ::= "SIZE" "("SizeConstraint")"

SizeConstraint     ::= SingleValue | Range

SingleValue        ::= value(INTEGER)

Range              ::= value(INTEGER) ".." value
                       (INTEGER)

IntegerConstraint  ::= Range

MatchTypes         ::= "MATCHES FOR" Matches | empty

Matches            ::= Match Matches | Match

Match              ::= "EQUALITY" | "SUBSTRINGS" |
                           "ORDERING"

END
```

§ 9.5.1 节列举的定义的各部分与宏引入的各种符号之间的相应关系如下：

a) 指定给属性类型的客体标识符是宏 MACRO 值分配提供的值；

b) 属性类型的属性语法是 **AttributeSyntax** 乘积给出的。这指的是个别定义的属性语法，或者通过提供其 ASN.1 类型和匹配规则（见 § 9.6），以明显方式定义一个属性语法。若采用个别定义语法，则可以选择指出特定串类型的长短限制或指出特定整数类型的值的范围；

c) 倘若 **MultiValued** 的乘积为单值 "**SINGLE VALUE**"，则属性亦为单值；倘若 **MultiValued** 的乘积为 "**MULTIVALUE**" 或空，则该属性可有一个或多个值。

注 — 在建议 X.520 中，宏用来定义被选用的属性类型。

倘若选择了 "**empty**" 类型符号，则结果符号（"**ATTRIBUTE**"）可用来指出任意可能的属性类型。

9.5.4 在 § 7.3.3 中指出的属性类型，即本号码簿所掌握的和为自己使用的属性类型的定义如下：

```
ObjectClass ::= ATTRIBUTE
     WITH ATTRIBUTE-SYNTAX objectIdentifierSyntax

AliasedObjectName ::= ATTRIBUTE
     WITH ATTRIBUTE-SYNTAX distinguishedNameSyntax
     SINGLE VALUE
```

注 1 — 这是偏定义：在建议 X.520 中，客体标识符实际上是被指定给这些属性类型，以便在本建议系列中提供这些客体标识符的单分配点。

注 2 — 在这些定义中提到的属性语法本身在 § 9.6.5 中定义。

9.6 属性语法定义

9.6.1 属性语法定义包括：

a) 任选项：给属性语法指定一个客体标识符；

b) 在 ASN.1 的属性语法中指出数据类型；

c) 为一个提出的值和 DIB 中保存的目标属性值之间的匹配定义适当的规则。下列匹配规则中的一部分或全部可以为某一特定属性语法下定义：

i) 等同：适用于任何属性语法。给出的值必须符合于该属性语法的数据类型；

ii) 子串：适用于具有串数据类型的任何属性语法。给出的值必须是一个序列（**"SEQUENCE OF"**），它的每一个元素都与数据类型相符；

iii) 顺序：适用于任何属性语法，只要这个语法有一个规则，允许把给出的值描写为小于、等于、或大于一个目标值。给出的值必须与属性语法的数据类型相符。

9.6.2 倘若未定义等同匹配规则，则号码簿：

a) 将此属性语法的属性值视为具有 **ANY**（**任意**）类型，即号码簿不检查该值是否与该属性语法给出的数据类型相符；

b) 将不试探为给出的值寻找相匹配的目标属性值。

注 — 因此该号码簿将不允许那样的属性使用于可识别名或修改某一特定值。

9.6.3 倘若定义了等同匹配规则，则号码簿：

a) 将此属性语法的属性值视为具有"任何为该属性语法给出的数据类型定义的"**ANY DEFINED BY** 类型；

b) 将只根据为该属性语法定义的匹配规则进行匹配；

c) 将只对 §9.6.1 c）所述的适当数据类型给出的值进行匹配。

9.6.4 下列 ASN.1 宏可以（但不必须）用来定义属性语法：

```
ATTRIBUTE-SYNTAX MACRO ::=
BEGIN
TYPE NOTATION ::=  Syntax
                   MatchTypes | empty
VALUE NOTATION ::=
     value (VALUE OBJECT IDENTIFIER)
Syntax ::=  type
MatchTypes ::= "MATCHES FOR" Matches | empty
Matches ::= Match Matches | Match
Match ::=  "EQUALITY" | "SUBSTRINGS" | "ORDERING"
END
```

§9.6.1 所述定义的各部分与宏引入的各种符号之间的对应关系如下：

a) 指定给属性语法的客体标识符是宏在值分配过程中提供的值；

b) 属性语法的数据类型是语法（**Syntax**）乘积给出的数据类型，即跟随宏名的那个数据类型；

c) 倘若“**EQUALITY**”、“**SUBSTRINGS**”或“**ORDERING**”出现在 **MatchTypes** 乘积中，则定义的匹配规则分别为“等同”、“子串”或“顺序”。倘若乘积为空，则表示没有匹配规则。

倘若选择了“**empty**”符号，则结果符号（“**ATTRIBUTE-SYNTAX**”）可用来指定任何属性语法。

注 1 — 在宏里不提供对定义匹配规则的支持。为此必须采用自然语言或其他手段。

注 2 — 宏在建议 X.520 中是用来定义选择的属性语法。

9.6.5 在§9.5.4 使用的属性语法在§9.6.5.1 和§9.6.5.2 中定义。

注 — 这些都是偏定义：在建议 X.520 中，客体标识符实际被分配给这些属性语法，以便在本建议序列中单点分配这些客体标识符。

9.6.5.1 ObjectIdentifierSyntax 定义如下：

```
ObjectIdentifierSyntax ::=
    ATTRIBUTE-SYNTAX
        OBJECT IDENTIFIER
        MATCHES FOR EQUALITY
```

等同匹配规则在 ASN.1 类型客体标识符的定义中是固有的。

9.6.5.2 DistinguishedNameSyntax 定义如下：

```
DistinguishedNameSyntax ::=
    ATTRIBUTE-SYNTAX
        DistinguishedName
        MATCHES FOR EQUALITY
```

当且仅当下列事件均为真时，给出的可识别名值才与目标可识别名的值相等：

a) 每一个可识别名的 RDN 数相等；

b) 每个对应 RDN 的 AVA 数相等；

c) 对应 AVA（具有相同属性类型）具有等同匹配的属性值（在这样的匹配情况下，属性值，不论是给出值还是目标值，和包含它们的可识别名一样，在总匹配情况下所起的作用相同。）

第 3 部分 — 保密模型

10 保密

10.1 号码簿存在于各种管理机构提供访问号码簿 DIB 片段的环境之中。这样的访问必须符合 DIB 片段所在保密领域内的保密政策（见建议 X.509）。

10.2 这里将讨论有关保密政策的两个组成部分：

a) 授权政策的定义；

b) 验证政策的定义；

10.3 在本号码簿里，授权的定义包括下列方法：

a） 访问权的说明；

b） 实施访问权（访问控制）的方法；

c） 维护访问权。

10.4 在本号码簿中，验证的定义包括验证下列身份的方法：

a） 验证 DSA 和号码簿用户身份证；

b） 验证在访问点上收到的那一信息的发送者身份证。

收到的信息的完整性属于本地决策，且必须符合现行保密政策。

10.5 本建议未定义保密政策。

10.6 附件 F 提供描述访问权的指南。

10.7 建议 X.509 定义验证规程。DAP 和 DSP 可通过签署申请分别提供对发起者的强验证和该申请的数据完好性，以及通过签署结果来提供响应者的强验证和该结果的数据完好性。DAP 可提供 DUA 和 DSA 之间的简单验证。DSP 可提供两个 DSA 之间的简单验证。

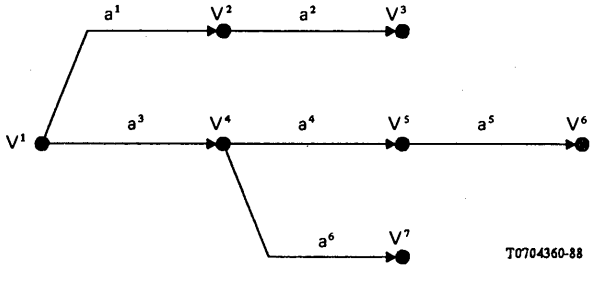
10.8 使用号码簿的应用行政权威可以采用他们自己的保密政策。号码簿能够通过掌握通信实体的验证信息（例如可识别名、口令、证件）来支持应用。这方面的问题将在建议 X.509 中进一步阐述。

附 件 A

（附于建议 X.501）

树的数学规则

本附件不属于本标准的一部分。



一个树是由一组点和一组有方向性的线构成的；前者叫*极点*，后者叫*弧线*。每一弧线 a 由极点 V 起至极点 V′ 止。例如，上图的树有七个极点，标以 V^1 至 V^7，和六根弧线，标以 a^1 至 a^6。

两个极点 V 和 V′ 分别叫做 V 至 V′ 弧线 a 的*起始*和*最终极点*。例如，V^2 和 V^3 分别为弧线 a^2 的起始和最终极点。几根不同的弧线可以有相同的起始极点，但不能有相同的最终极点。如弧线 a^1 和 a^3 具有相同的起始极点 V^1，但图中没有具有相同最终极点的两个弧线。

不是任何弧线的最终极点的极点通常叫做根极点，或不规范地称为树“根”。例如，图中 V^1 是根。

不是任何弧线的起始极点的极点通常不规范地称为叶极点，或不规范地被称为树形图的“叶子”。例如，极点 V^3，V^6 和 V^7 都是叶子。

由极点 V 至 V′ 的一条有方向性的路径是一组弧线$(a^1,a^2,\cdots,a^n)(n\geqslant 1)$，使得 V 成为弧线 a^1 的起始极点，V′ 成为弧线 a^n 的一个最终极点，而弧线 a^k 的最终极点也是弧线 a^{k+1}的起始极点，其中 $1\leqslant k<n$。例如，由 V^1 至 V^6 有方向性路径是一组弧线(a^3,a^4,a^5)。“路径”一词应理解为是由根至极点的有方向性的路径。

附 件 B

（附于建议 X.501）

客体标识符的用法

本附件是本标准的一部分。

本附件阐述客体标识符子树的上部，即存放本建议系列指定的所有客体标识符的空间。为此目的，本附件提供叫做“有用定义”（“**UsefulDefinitions**”）的 ASN.1 模块，在该模块中子树里的所有非叶节点均有指定名称。

```
UsefulDefinitions     {joint-iso-ccitt ds(5) modules(1)
                      usefulDefinitions(0)}
DEFINITIONS ::=
BEGIN

EXPORTS
     module, serviceElement, applicationContext, attributeType, attributeSyntax, objectClass,
     algorithm, abstractSyntax, attributeSet,

     usefulDefinitions, informationFramework, directoryAbstractService,
     directoryObjectIdentifiers, algorithmObjectIdentifiers, distributedOperations,
     protocolObjectIdentifiers, selectedAttributeTypes, selectedObjectClasses,
     authenticationFramework, upperBounds,
     dap,dsp

     id-ac, id-ase, id-as, id-ot, id-pt;

ds              OBJECT IDENTIFIER ::= {joint-iso-ccitt ds(5)}

-- categories of information object --

module              OBJECT IDENTIFIER ::= {ds 1}
serviceElement      OBJECT IDENTIFIER ::= {ds 2}
applicationContext  OBJECT IDENTIFIER ::= {ds 3}
attributeType       OBJECT IDENTIFIER ::= {ds 4}
attributeSyntax     OBJECT IDENTIFIER ::= {ds 5}
objectClass         OBJECT IDENTIFIER ::= {ds 6}
attributeSet        OBJECT IDENTIFIER ::= {ds 7}
algorithm           OBJECT IDENTIFIER ::= {ds 8}
abstractSyntax      OBJECT IDENTIFIER ::= {ds 9}
object              OBJECT IDENTIFIER ::= {ds 10}
port                OBJECT IDENTIFIER ::= {ds 11}
```

```
-- modules --

usefulDefinitions                    OBJECT IDENTIFIER ::= {module 0}
informationFramework                 OBJECT IDENTIFIER ::= {module 1}
directoryAbstractService             OBJECT IDENTIFIER ::= {module 2}
distributedOperations                OBJECT IDENTIFIER ::= {module 3}
protocolObjectIdentifier             OBJECT IDENTIFIER ::= {module 4}
selectedAttributeTypes               OBJECT IDENTIFIER ::= {module 5}
selectedObjectClasses                OBJECT IDENTIFIER ::= {module 6}
authenticationFramework              OBJECT IDENTIFIER ::= {module 7}
algorithmObjectIdentifiers           OBJECT IDENTIFIER ::= {module 8}
directoryObjectIdentifiers           OBJECT IDENTIFIER ::= {module 9}
upperBounds                          OBJECT IDENTIFIER ::= {module 10}
dap                                  OBJECT IDENTIFIER ::= {module 11}
dsp                                  OBJECT IDENTIFIER ::= {module 12}
distributedDirectoryObjectIdentifiers OBJECT IDENTIFIER ::= {module 13}

-- synonyms --

id-ac     OBJECT IDENTIFIER ::= applicationContext
id-ase    OBJECT IDENTIFIER ::= serviceElement
id-as     OBJECT IDENTIFIER ::= abstractSyntax
id-ot     OBJECT IDENTIFIER ::= object
id-pt     OBJECT IDENTIFIER ::= port

END
```

附 件 C

（附于建议 X.501）

ASN.1 中的信息框架

本附件是本建议的一部分。

本附件提供本建议中的所有 ASN.1 类型、值和宏定义的摘要。本定义构成 ASN.1“信息框架”（“InformationFramework”）模块。

```
InformationFramework    {joint-iso-ccitt ds(5) modules(1)
                         informationFramework(1)}
DEFINITIONS ::=
BEGIN

EXPORTS
    Attribute, AttributeType, AttributeValue, AttributeValueAssertion,
    DistinguishedName, Name, RelativeDistinguishedName,
    OBJECT-CLASS,ATTRIBUTE,ATTRIBUTE-SET,ATTRIBUTE-SYNTAX,
    Top, Alias,
    ObjectClass, AliasedObjectName,
    ObjectIdentifierSyntax, DistinguishedNameSyntax;

IMPORTS
    selectedAttributeTypes, selectedObjectClasses
        FROM    UsefulDefinitions {joint-iso-ccitt ds(5) modules(1)
                usefulDefinitions(0)}
    top
        FROM SelectedObjectClasses selectedObjectClasses
    objectIdentifierSyntax, distinguishedNameSyntax, objectClass, aliasedObjectName
        FROM SelectedAttributeTypes selectedAttributeTypes;
```

```
-- attribute data types --

Attribute                ::= SEQUENCE{
                                 type          AttributeType
                                 values SET OF AttributeValue
                                 -- at least one value is required --}

AttributeType            ::= OBJECT IDENTIFIER

AttributeValue           ::= ANY

AttributeValueAssertion  ::= SEQUENCE {AttributeType, AttributeValue}

-- naming data types --

Name          ::=   CHOICE {-- only one possibility for now --
                        RDNSequence}

RDNSequence                  ::=   SEQUENCE OF
                                       RelativeDistinguishedName

DistinguishedName            ::=   RDNSequence

RelativeDistinguishedName    ::=   SET OF AttributeValueAssertion

-- macros --

OBJECT-CLASS MACRO   ::=
BEGIN

    TYPENOTATION         ::= SubclassOf MandatoryAttributes
                                 OptionalAttributes
    VALUENOTATION        ::= value (VALUE OBJECT IDENTIFIER)

    SubclassOf           ::= "SUBCLASS OF" Subclasses | empty
    Subclasses           ::= Subclass | Subclass "," Subclasses
    Subclass             ::= value (OBJECT-CLASS)
    MandatoryAttributes  ::= "MUST CONTAIN {"Attributes"}" | empty
    OptionalAttributes   ::= "MAY CONTAIN {"Attributes"}" | empty

    Attributes           ::= AttributeTerm | AttributeTerm ","
                             Attributes
    AttributeTerm        ::= Attribute | AttributeSet
    Attribute            ::= value(ATTRIBUTE)
    AttributeSet         ::= value(ATTRIBUTE-SET)

END

ATTRIBUTE-SET-MACRO ::=

    BEGIN

    TYPE NOTATION        ::= "CONTAINS" "{Attributes"}" | empty

    VALUE NOTATION       ::= value(VALUEOBJECTIDENTIFIER)

    Attributes           ::= AttributeTerm | AttributeTerm "," Attributes

    AttributeTerm        ::= Attribute | AttributeSet
    Attribute            ::= value(ATTRIBUTE)
    AttributeSet         ::= value(ATTRIBUTE-SET)

    END

ATTRIBUTE MACRO      ::=
BEGIN

TYPENOTATION         ::= AttributeSyntax Multivalued | empty
VALUENOTATION        ::= value(VALUE OBJECT IDENTIFIER)

AttributeSyntax      ::= "WITH ATTRIBUTE-SYNTAX"  SyntaxChoice

Multivalued          ::= "SINGLE VALUE" | "MULTI VALUE" | empty

SyntaxChoice         ::= value(ATTRIBUTE-SYNTAX)
                         Constraint | type Match Types
```

```
Constraint                  ::=  "("ConstraintAlternative")" | empty

ConstraintAlternative       ::=  StringConstraint | IntegerConstraint
StringConstraint            ::=  "SIZE" "("SizeConstraint")"
SizeConstraint              ::=  SingleValue | Range
SingleValue                 ::=  value(INTEGER)
Range                       ::=  value(INTEGER) ".." value(INTEGER)
IntegerConstraint           ::=  Range
MatchTypes                  ::=  "MATCHES FOR" Matches | empty
Matches                     ::=  Match Matches | Match
Match                       ::=  "EQUALITY" | "SUBSTRINGS" | "ORDERING"
END

ATTRIBUTE-SYNTAX MACRO ::=
BEGIN
    TYPENOTATION    ::=  Syntax MatchTypes | empty
    VALUENOTATION   ::=  value(VALUE OBJECT IDENTIFIER)

    Syntax          ::=  type

    MatchTypes      ::=  "MATCHES FOR "Matches | empty
    Matches         ::=  Match Matches | Match
    Match           ::=  "EQUALITY" | "SUBSTRINGS" | "ORDERING"
END

-- object classes --

Top      ::=     OBJECT-CLASS
                     MUST CONTAIN {objectClass}
Alias    ::=     OBJECT-CLASS
                     SUBCLASS OF top
                     MUST CONTAIN {aliasedObjectName}

-- attribute types --

ObjectClass      ::= ATTRIBUTE
                     WITH ATTRIBUTE-SYNTAX objectIdentifierSyntax
AliasedObjectName ::= ATTRIBUTE
                     WITH ATTRIBUTE-SYNTAX distinguishedNameSyntax
                     SINGLE VALUE

-- attribute syntaxes --

ObjectIdentifierSyntax ::=
    ATTRIBUTE-SYNTAX
    OBJECT IDENTIFIER
    MATCHES FOR EQUALITY
DistinguishedNameSyntax ::=
    ATTRIBUTE-SYNTAX
    DistinguishedName
    MATCHES FOR EQUALITY
END
```

附 件 D

（附于建议 X.501）

定义索引表

本附件不是本建议的一部分。
本附件按英文字母列出本建议定义的所有术语和交叉参考，以及定义所在的章节号。

A access point 访问点 …… §5
Administration Directory Management Domain 主管部门号码簿管理域 …… §5
alias 别名 …… §8
alias entry 别名登录项 …… §6
attribute 属性 …… §7
attribute type 属性类型 …… §7
attribute value 属性值 …… §7
attribute value assertion 属性值判断语 …… §7

D the Directory 号码簿 …… §5
Directory entry 号码簿登录项 …… §6
Directory Information Base (DIB) 号码簿信息库 …… §6
Directory Information Tree (DIT) 号码簿信息树 …… §6
Directory Management Domain (DMD) 号码簿管理域 …… §5
Directory name 号码簿名 …… §8
Directory schema 号码簿体制 …… §9
Directory System Agent (DSA) 号码簿系统代理 …… §5
Directory User Agent (DUA) 号码簿用户代理 …… §5
distinguished name 可识别名 …… §8
DIT Structure rule DIT 结构规则 …… §9

E entry 登录项 …… §6

I immediate(ly) subordinate 直接下级 …… §6
immediate(ly) superior 直接上级 …… §6

N name 名称 …… §8
naming authority 命名权威 …… §8

O object (of interest)（感兴趣的）客体 …… §6
object class 客体类别 …… §6
object entry 客体登录项 …… §6

P Private Directory Management Domain 私营号码簿管理域 …… §5
purported name 声称名 …… §8

R relative distinguished name 相对可识别名 …… §8

S subordinate 下级 …… §6
superior 上级 …… §6

附 件 E

（附于建议 X.501）

名称设计准则

本附件不是本标准的一部分。

信息框架是通用的，并允许在DIT内存放各种登录项和属性。由于名称与通往DIT的路径密切相关，因此意味着名称的种类是任意的。本节提供名称设计中应考虑的一些准则。某些准则已在建议X.521中用于建议采用的名称格式的设计。在所建议的名称格式不适合于某些客体名设计的场合，也可采用本准则。

目前只讨论对用户友好的准则。

注 — 不要求所有的名称都对用户友好。

E.1 对用户友好

人们直接使用的名称必须对用户友好。对用户友好的名称是从人类用户的角度着眼而不是计算机。对用户友好的名称是指便于人们推论、记忆和理解的名称，而不是便于计算机翻译的名称。

对用户友好的目标可根据下面的两条原则更确切地加以描述：

— 在一般情况下，人总是能够根据他所自然掌握的信息正确地推测出客体对用户友好的名称。例如，通过一般的公事往来获得有关一位女士的情报，就能够猜出她的名字。

— 当客体名含糊不清时，号码簿应当能够发现而不应当下决论断言该名是指某一特定客体。例如，若两个人的姓相同，则仅仅知道姓是没有充分的根据来识别他们的。

对用户友好的目标有以下几个子目标：

a） 不应当因名称人为地消除自然存在的含糊性。例如，若有两个人共用一个姓“Jones”，则他们不应当对“WJones”或“Jones2”的称呼做出反应。相反地，应当使用惯用命名法做为识别两个实体的手段。例如，除姓之外，为了鉴别可能还需要提供名字和中间名的第一个字母。

b） 应当允许使用一般缩写名和拼音上的一般差别。例如，某人在Conway Steel Corporation公司任职，若他的名字后面附公司名，为了识别该单位，可从“Conway Steel Corporation”，“Conway Steel Corp”，“Conway Steel”和“CSC”中任选一个就足够。

c） 在某些情况下，为了对用户更友好或减少对某一个登录项的搜索范围，可采用别名进行搜索。下面是使用别名进行搜索的例子：图E-1/X.501中，在Osaka的公司可用名称{C=Japan，L=Osaka，O=ABC，OU=Osaka-branch}来识别。

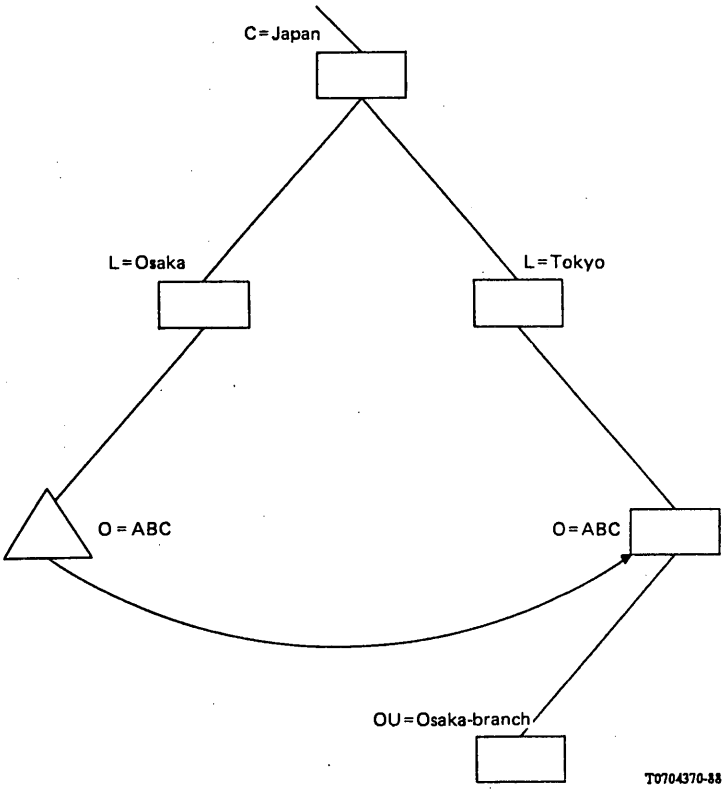


图 E-1/X.501

别名的用法举例

d) 倘若使用由几个部分组成的名称，则必备部分和可选部分应相对地短一些，以便于记忆。

e) 倘若使用由几个部分组成的名称，则其排列的秩序一般是无关紧要的。

f) 对用户友好的名称不应包含计算机地址。

附 件 F

（附于建议 X.501）

访 问 控 制

本附件不是本建议的一部分。

F.1 绪论

号码簿用户根据信息保护访问控制策略提供的访问控制权享有访问 DIB 信息的权力。在本建议系列中，访问控制是由本地决策负责。但应当指出，实现访问控制需要某种手段，本建议系列的新版本有可能

提供建立、维护和使用访问控制信息的标准化手段的定义。本附件说明访问控制的原则，并给出两种可能的访问控制方法。

F.2 原则

指导建立访问控制管理规程的两个原则是：

a) 必须提供保护号码簿信息的手段，防止局外人私自检测、查阅和修改这些信息，包括防止局外人私自修改 DIT；

b) 有关的 DSA 必须掌握鉴别用户控制权所需要的信息，避免仅仅为了鉴别用户控制权而在远端进行的操作。

F.3 保护项目

目前已提出的保护项目有以下几项：

a) 对 DIT 的整个子树的保护；

b) 对各别登录项的保护；

c) 对一个登录项的全部属性的保护；

d) 对属性值的被选实例的保护。

F.4 访问项目种类

预计至少需要五种访问项目。倘若不允许用户访问某一种受保护的访问项目，则号码簿在应答时尽可能声明该项目不存在。

表 F-1/X.501 列出访问项目种类，项目栏说明受保护的项目内容是登录项（E），或是属性（A），或二者皆是（EA）。

表 F-1/X.501

访问项目种类

种　类	项　目	说　明
检测	A	允许对受保护的项目进行检测
比较	A	允许对给出的值与受保护的项目进行比较
读	A	允许对受保护的项目进行读取
修改	A	允许对受保护的项目进行更新
增/删	EA	允许在受保护的项目内部建立和删除新组成部分（属性或属性值）
命名	E	允许建立和删除受保护登录项的直接下级登录项，和修改后者的相对可识别名

F.5 访问权的鉴别

访问控制管理方案对每一个受保护的项目以明显方式或隐含方式提供一张访问权表。在表格里成对地列出用户集合和对应的访问项目种类。

根据申请而获得的信息(不论是在BIND中提供的用户身份和证明文件,还是操作变量提供的信息)。应当能够检验某用户是否出现在一个（或多个）被指出的用户集合里。

至少存在下列两种可能性：

a) 用户集合是采用用户的可识别名描述的——例如采用用户的可识别名或上级的识别名，以及一个说明包括整个子树的标志；

b) 用户集合只提供一个功能，并且隐含地包括了拥有这种功能的所有用户。这个方案要求在本地提供上述用户功能，否则应在BIND或操作变量中提供。后者可能要求对目前定义的协议做一些补充。

建　议　X.509

号码簿验证框架①

(1988年订于墨尔本)

目　录

0 绪论…… 49

1 应用范围…… 50

2 参考文献…… 50

3 定义…… 51

4 符号与缩写词…… 52

第1部分 — 简单验证 …… 54

5 简单验证规程…… 54

第2部分 — 强验证 …… 56

6 强验证基础…… 56

7 如何获得用户的公用钥匙…… 57

① 建议X.509与ISO 9594-8，信息处理系统—开放系统互连—号码簿验证框架的制定是在紧密合作下进行的，而且在技术上是协调的。

8 数字签名 …… 61

9 强验证规程 …… 64

10 钥匙和证件的管理 …… 67

附件 *A* — 安全保密要求 …… 69

附件 *B* — 公用钥匙加密技术简介 …… 72

附件 *C* — *RSA* 公用钥匙密码系统 …… 74

附件 *D* — 散列函数 …… 76

附件 *E* — 利用强验证法保护数据不受侵犯 …… 77

附件 *F* — 数据机密性 …… 78

附件 *G* — 使用 *ASN.1* 描述的验证框架 …… 78

附件 *H* — 算法客体标识符的参考定义 …… 81

0 绪论

0.1 本文和本系列其他文件的目的在于简化信息处理系统之间的互连，以便提供号码簿服务。全套系统，包括它们的号码簿信息，可视为一个整体，称为号码簿。号码簿包含的全部信息称为码簿信息库（DIB），其典型用途是简化客体之间以及同这些客体、或围绕这些客体（如 OSI 应用实体、人员、终端和分布表）的通信联系。

0.2 号码簿在开放系统互连中扮演重要角色，开放系统的目的是除使用互连标准之外设法只通过很少的技术协议就能实现下列信息处理系统之间的互连：

— 不同厂家；
— 不同管理部门；
— 不同复杂程度；
— 不同年代。

0.3 有许多应用提出保护信息通信不受侵犯的保密要求。附件 A 扼要地介绍一些常见的威胁和可用来保护信息通信的保密服务和机制。实质上，所有的保密服务均取决于可靠地识别参与通信的各方使用的身份证，即验证。

0.4 本建议定义号码簿向用户提供验证服务的框架、上述用户包括号码簿本身以及其他应用和服务。由于号码簿是参与通信的各方获得验证信息（即验证基础的知识）固有的地方，号码簿是能够满足用户对验证和其他保密服务要求的。因为号码簿保存通信所需要的以及通信之前应提供的其他信息，所以号码簿是存放验证信息的固有地方。利用这种方法从号码簿中获得对方用户的验证信息是同获得地址类同的。由于号码簿在通信中应用很广，可以预料本验证框架将广泛应用于许多领域。

1 应用范围

1.1 本建议：

— 规定号码簿验证信息的格式；
— 说明如何从号码簿里获得验证信息；
— 假设如何在号码簿里组织和存放验证信息的方法；
— 定义使用验证信息进行验证的三种方法，并说明本验证如何支持其他保密服务。

1.2 本建议说明两种验证等级：简单验证和强验证，前者使用口令来验证身份，后者使用密码技术提供凭证。简单验证对非法访问只能提供有限的保护，只有采用强验证才能获得保密和安全服务。上述方法不是为了建立一个一般性的验证框架；但若认为上述验证技术已经足够，当然它也适用于一般用途。

1.3 验证（以及其他安全保密服务）只能在具体的安全保密政策范围内提供。特定应用的用户应当制定自己的安全保密政策，但可能要受到标准所提供的服务的约束。

1.4 为了能够根据由号码簿获得的验证信息来达到验证目的，需要交换协议。因此，为使用验证框架的某些应用下定义的标准应当包括有关这种协议交换的描述。在某些应用中，为了从号码簿里获得凭证常使用建议 X.519 所述的号码簿访问协议（DAP）。

1.5 本建议所述的强验证方法采用公用钥匙密码系统。该密码系统的主要优点在于用户证件可做为属性保存在号码簿里，并可在号码簿系统内自由传递，而且号码簿用户可采用获得其他号码簿信息的同样方式来获得证件。用户证件是用离线方式编造的，并由编造者负责存放在号码簿里。用户证件是由离线的发证机构生成的，该机构同号码簿里的 DSA 是完全分开的。特别是，不存在向号码簿提供者提出安全贮存或传递用户证件的任何特殊要求。

有关公用钥匙密码术的简单介绍在附件 B 中描述。

1.6 在一般情况下，只要验证框架具备 § 6.1 所述的特性，则验证框架与具体的密码算法无关。可使用的算法有很多，但是，希望验证的两个用户必须支持相同的密码算法以保证正确的验证。因此，对一组相关应用而言，选择单一的算法是为了尽可能扩大能够进行安全验证和通信的用户数。附件 C 列举公用钥匙密码算法的一个实例。

1.7 同理，希望进行验证的两个用户必须支持相同的散列函数（见 § 3.3 f)）（用于编造凭证和验证牌）。此外，原则上有许多散列函数可供选择，其代价是减少能够参与验证的用户数。附件 D 简单介绍散列函数及其例子。

2 参考文献

2.1 ISO 7498-2：信息处理系统—开放系统互连—保密结构。

3 定义

3.1 本建议使用OSI参考模型第2部分关于安全保密措施的有关术语如下：

a) *asymmetric (encipherment)* 不对称（加密）

b) *authentication exchange* 验证交换

c) *authentication information* 验证信息

d) *confidentiality* 机密

e) *credentials* 凭证

f) *cryptography* 密码术

g) *data origin authentication* 数据源验证

h) *decipherment* 译码

i) *encipherment* 加密

j) *key* 钥匙

k) *password* 口令

l) *peer-entity authentication* 对等实体验证

m) *symmetric (encipherment)* 对称（加密）

3.2 下列术语在建议X.501中定义：

a) *attribute* 属性

b) *Directory Information Base* 号码簿信息库

c) *Directory Information Tree* 号码簿信息树

d) *Distinguished name* 可识别名

e) *entry* 登录项

f) *object* 客体

g) *root* 根

3.3 下列术语在本建议中定义和使用：

a) *authentication token (token)* 验证牌（令牌）：在强验证交换的过程中传递的信息，它可用来验证发送者的身份。

b) *user certificate (certificate)* 用户证件（证件）：用户的公用钥匙和其他一些信息，使用发证机构的密钥进行加密，使局外人无法伪造。

c) *certification authority* 发证机构：一个或多个用户所信赖的权力机构；该机构负责编造和颁发证件。

d) *certification path* 验证路径：在DIT中，客体证件的一个有序序列，它和路径上的初始客体的公用钥匙均可处理，以获得路径上最终客体的公用钥匙。

e) *cryptographic system*，*cryptosystem* 密码系统：从普通文件到加密文件的一系列变换和逆变换；拟使用的特定变换可用钥匙选择。变换通常是采用数学算法定义的。

ARCHIVES U.I.T. GENEVE

f) *hash function* 散列函数：一种数学函数，它把大域（可能很大）中的值映射到较小的范围。“好的”散列函数是这样的：将该函数应用于域中的一（大）群值的结果将均匀地（显然是随机的）分布于整个范围。

g) *one-way function* 单向函数：一个（数学）函数 f，它很容易运算，但若给出范围以内的某一个值 y，则要在域中找到一个值 x，使得 f（x）=y 是相当困难的。y 可能拥有几个值，使得不难找到合适的 x 值。

h) *public key* 公用钥匙：（在一个公用钥匙密码系统里）公开的用户钥匙对中的一把钥匙。

i) *private key* 专用钥匙（密钥—不赞成的）：（在一个公用钥匙密码系统里）用户钥匙对中的一把钥匙，只有该用户知道这把钥匙。

j) *simple authentication* 简单验证：使用简单口令进行验证。

k) *security policy* 安全保密政策：由保密机构制定的一套规章；该保密机构对安全保密服务以及设施的使用和提供进行监督。

l) *strong authentication* 强验证：采用密码凭证的验证方式。

m) *trust* 信任：在一般情况下，当某一个实体假设另一个实体表现的行为完全合乎它的期望时，就可以认为该实体（第一个）“信任”第二个实体。这样信任可能只适用于某一个特定功能。在验证框架里，信任的主要作用是说明验证实体与发证机构之件的关系；验证实体必须保证发证机构编造的证件是有效而且可靠，从而可以信任。

n) *certificate serial number* 证件系列号：一个整数值，它在颁发证件的 CA 内是唯一的，在 CA 颁发的证件中是单值。

4 符号与缩写词

4.1 本建议使用的符号在表 1/X.509 中定义。

注 — 表中，符号 X，X_1，X_2 等代表用户名，I 代表任意信息。

4.2 本建议使用下列缩写词：

CA 发证机构
DIB 号码簿信息库
DIT 号码簿信息树
PKCS 公用钥匙密码系统

表 1/X.509

符　号

符　号	含　义
Xp	用户 X 的公用钥匙
Xs	X 的密钥
Xp [I]	使用 X 的公用钥匙对信息 I 进行加密
Xs [I]	使用 X 的密钥对 I 进行加密
X {I}	用户 X 签署 I。它包括 I 和附加的加密摘录
CA (X)	用户 X 的发证机构
CA^n (X)	(其中，n>1)；CA (CA (…n 次… (X)))
$X_1 \ll X_2 \gg$	用户 X_2 的证件是由发证机构 X_1 颁发的
$X_1 \ll X_2 \gg X_2 \ll X_3 \gg$	证件链（任意长度），其中的每一项代表发证机构的证件；该发证机构产生下一个证件。它在功能上等效于证件 $X_1 \ll X_{n+1} \gg$。例如，证件 A≪B≫B≪C≫与证件 A≪C≫是等效的，即给出 Ap 能够找到 Cp。
$X_{1p} \cdot X_1 \ll X_2 \gg$	打开证件（或证件链）以便取出公用钥匙的操作。它是一个插入运算符，左边的操作数是发证机构颁发的证件。操作的结果是用户的公用钥匙；右边的操作数是发证机构颁发给该用户的证件。例如：Ap・A≪B≫B≪C≫表示使用公用钥匙 A；从 B 的证件中取出 B 的公用钥匙 Bp；然后使用 Bp 打开 C 的证件。上述操作的结果是获得 C 的公用钥匙 Cp。
A→B	由 A 至 B 的证件路径，它是由一连串的证件组成的，开头为 CA (A) ≪CA^2 (A) ≫，结尾为 CA (B) ≪B≫。

5 简单验证规程

5.1 简单验证的目的是根据用户的可识别名和双方同意（任选）的口令，以及双方对在一个领域内所使用和处理口令的手段的理解，来提供本地验证。简单验证主要是应用于本地，即应用于一个 DUA 和一个 DSA 之间，或一个 DSA 和另一个 DSA 之间进行的对等实体的验证。简单验证可通过以下几种手段实现：

a） 将用户的可识别名和（任选）明码口令（即无保护的口令）送到收信方去检验；

b） 传递用户的可识别名、口令、随机数和/或时戳；这些信息均采用单向函数加以保护；

c） 将 b)所述的受保护信息，连同随机数和/或时戳传递给对方；这些信息均采用单向函数加以保护。

注 1 — 不要求使用不相同的单向函数。

注 2 — 保护口令使用的信令规程可能将做为本文的扩展部分。

5.2 在口令不受保护的情况下，为了防止无权用户获得口令，必须提供最低限度的保护措施。不能认为这是保密服务的基础。保护用户的可识别名和口令可提高保密度。保护机制使用的典型算法通常是采用易于实现的不加密单向函数。

5.3 简单验证使用的一般规程如图 1/X.509 所示。

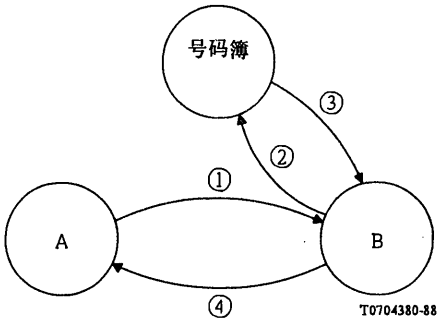


图 1/X.509

无保护的简单验证规程

5.3.1 简单验证的步骤如下：

1） 主叫用户 A 把自己的可识别名和口令送到被叫用户 B；

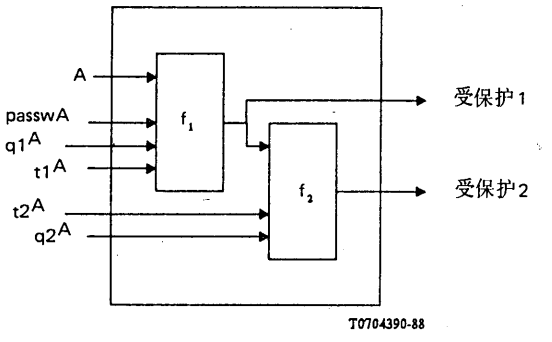
2） B 把 A 声称的可识别名和口令送到号码簿，后者把该口令同 A 在号码簿登录项中的用户口令属性进行比较（使用号码簿的比较操作）；

3） 号码簿向 B 证实（或否定）该凭证有效；

4） 可把验证成功（或失败）的结果向 A 传递。

5.3.2 简单验证最基本的形式只包括步骤 1)，当 B 检验了可识别名和口令之后还可以包括步骤 4)。

5.4 图 2/X.509 表示生成加密识别信息的两种方法。其中，f_1 和 f_2 代表单向函数（相同或不相同）；时戳和随机数是任选的并由双方协商确定。



A = 用户可识别名

t^A = 时戳

$passw^A$ = A的口令

q^A = 随机数码，包括任选计数器

图 2/X.509

受保护的简单验证

5.4.1 图 3/X.509 表示受保护的简单验证规程。

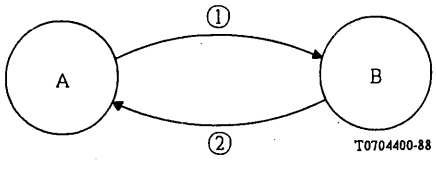


图 3/X.509

受保护的简单验证规程

受保护的简单验证的步骤如下（开始只使用 f_1）：

1) 主叫用户 A 把自己的加密识别信息（验证者 1）送到用户 B。采用图 2/X.509 的单向函数（f_1）来达到保护目的，其中时戳和/或随机数（若使用）是为了尽量减少重放次数和为了隐藏口令的目的而设置的。

受保护的 A 口令可表示如下：

$$Protected1 = f1\ (t1^A,\ q1^A,\ passw\ A)。$$

送到 B 的信息可表示如下：

$$Authenticator\ 1 = t1^A,\ q1^A,\ A,\ Protected\ 1。$$

B 通过生成 A 在本地受保护的口令（Protected 1）的拷贝（使用 A 所提供的时戳、可识别名和任选的附加时戳和/或随机数以及 A 口令在本地的拷贝）对 A 提供的加密识别信息进行检验。B 把 A 声称的识别信息（Protected 1）同在本地生成的值相比较（检查是否相等）。

2） B 向 A 证实（或否定）加密识别信息的验证结果。

5.4.2 为了获得更高的保密性能，可以（使用 f_1 和 f_2）对 § 5.4.1 所述的规程进行修改。其主要区别如下：

1） A 把自己的（附加）加密识别信息（Authenticator 2）送到 B。为了提高保密性能，采用单向函数 f_2，如图 2/X.509 所示。保密性能的改善可表示如下：

Protected 2 $=f_2$（$t2^A$，$q2^A$，Protected 1）。

送到 B 的信息可表示如下：

Authenticator 2 $=t1^A$，$t2^A$，$q1^A$，$q2^A$，Protected 2。

为了比较，B 生成 A 的附加保密口令的本地值，并把它同 Protected 2 的本地值相比较（检查是否相等）。原则上与 § 5.4.1 的步骤 1）类同。

2） B 向 A 证实（或否定）加密识别信息的验证结果。

注 — 本节定义的规程是以 A 和 B 描述的。应用于号码簿时（在建议 X.511 和 X.518 中描述），A 可能是一个同 DSA，B 耦合的 DUA；或者 A 是一个同另一个 DSA，B 耦合的 DSA。

5.5 用户口令属性类型包含一个客体的口令。用户口令的属性值是由客体描述的一个串。

```
UserPassword ::=  ATTRIBUTE
            WITH  ATTRIBUTE-SYNTAX
                  OCTET STRING (SIZE (0..ub-user-password))
                  MATCHES FOR EQUALITY
```

5.6 下列 ASN.1 宏可用来定义因对某一特定数据类型使用单向函数而产生的另一种数据类型。

```
PROTECTED MACRO ::= SIGNATURE
```

第 2 部分 — 强验证

6 强验证基础

6.1 本建议采用的强验证法使用称为公用钥匙密码系统（PKCS）的一系列特性。这些密码系统（不对称的）包括一对钥匙，一个保密，另一个公开，而传统密码系统只使用一把钥匙。附件 B 简单介绍这些密码系统和使它们在验证中发挥作用的一些特性。要使 PKCS 在本验证框架中发挥作用，目前它必须具备这样的特性，即两种钥匙均可用于加密：若使用公用钥匙进行加密，则应采用密钥译码；反过来，若使用密码进行加密，则应使用公用钥匙译码。换句话说，$X_p \cdot X_s = X_s \cdot X_p$；其中，$X_p/X_s$ 是使用用户 X 的明钥/密钥的编码/译码函数。

注 — 其他 PKCS 类型不要求具备可置换特性，而且提供支持时也不必对本建议做大量的修改，这些类型可能成为将来的扩展类型。

6.2 本验证框架不硬性规定使用某一特定密码系统。本框架的目的是可适应任何合适的公用钥匙密码系统，从而能够支持未来的更先进的密码系统、数学技术或计算能力所带来的变化。但是，为了确保验证结果的可靠性，两个用户必须支持相同的密码算法。因此，就一组相关应用而言，选择单一的算法有助于扩大能参与可靠验证和通信的用户范围。附件 C 举了一个密码算法的例子。

6.3 验证的依据是每一个用户拥有一个特殊的可识别名。命名权威有责任分配可识别名。因此，每一个用户都应相信命名权威不会颁发重复的可识别名。

6.4 每一个用户是通过他拥有的密钥被识别的。第二个用户能够确定他的通信伙伴是否拥有密钥，并使用这一点来证实他的通信伙伴确实是该合法用户。证实结果是否有效取决于该密钥是否仍然只有该通信伙伴知道。

6.5 为了使用户能够确定他的通信伙伴是否拥有另一个用户的密钥，该用户本身必须拥有另一个用户的公用钥匙。从号码簿中的用户登录项中获得这个公用钥匙的值是轻而易举的事，但要检验它是否正确则较困难。检验公用钥匙的值有许多方法：§7 描述如何参照号码簿来检验用户的公用钥匙的方法。只有当号码簿里要求验证的两个用户之间有一条不间断的信任点链时，这个方法才起作用。这样的链可通过寻找公共信任点来建立。这个公共信任点必须通过一条不间断的信任点链与各个用户相连。

7 如何获得用户的公用钥匙

7.1 为了使用户信任验证规程，该用户必须从一个它所信任的机构那里获得其他用户的公用钥匙。这样的机构称为发证机构（CA），这个机构使用公用钥匙算法去证明该公用钥匙，并出示证件。该证件（在 §7.2 中描述）具有下列特性：

— 任何用户，凡是拥有从发证机构那里获得公用钥匙的访问权，均能够获得已被证实的公用钥匙；

— 除了发证机构之外，其他用户都不能修改证件，否则一定将被发现（即证件是不能伪造的）。

由于证件是不能伪造的，因此可把它们公布于号码簿而不需要特殊保护。

注 — 虽然各 CA 在 DIT 中是以可识别名明确定义的，但这并不意味着各 CA 的组织和 DIT 之间有任何关系。

7.2 发证机构通过签署（见 §8）一批信息，包括用户的可识别名和公用钥匙，出示证件。例如，设有一个用户，其可识别名为 A，则发证机构 CA 为该用户出示的证件可表示如下：

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

其中，SN 为证件系列号，AI 为签署证件使用的算法标识符，T^A 指示证件的有效期，它包括前后两个日期，表示证件的有效期限。由于假设 T^A 将在不少于 24 小时的周期内变化，预期系统将采用通用标准协调时间（UTC）做为基准时间。任何用户均可使用 CAP 的知识去检验证件上的签名。下列 ASN.1 数据类型可用来表示证件。

```
Certificate ::=        SIGNED SEQUENCE{
     version                  [0]Version DEFAULT 1988,
     serialNumber             SerialNumber,
     signature                Algorithmidentifier
     issuer                   Name
     validity                 Validity,
     subject                  Name,
     subjectPublicKeyInfo     SubjectPublicKeyInfo}

Version        ::=   INTEGER { 1988(0)}
SerialNumber ::=     INTEGER

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

SubjectPublicKeyInfo   ::=
      SEQUENCE{
            algorithm        AlgorithmIdentifier
            subjectKey       BIT STRING}

AlgorithmIdentifier      ::=
      SEQUENCE{
            algorithm        OBJECT IDENTIFIER
            parameters       ANY DEFINED BY algorithm
                                OPTIONAL}
```

7.3 参与强验证的每一个用户 A 的号码簿登录项包含 A 的证件。该证件是由 DIT 的一个实体——A 的发证机构产生的。A 的发证机构可能不是唯一的，它可用 CA（A）表示，或已明确一定是指 A 的发证机构时，可简单表示为 CA。这样以来，凡是知道 CA 的公用钥匙的任何用户都能找到 A 的公用钥匙。因此，寻找公用钥匙是一种循环的过程。

7.4 倘若试图获得用户 B 的证件的用户 A 已经获得了公用钥匙 CA（B），则这个过程就算完结。为了使 A 能够获得公用钥匙 CA（B），在每一个发证机构 X 的号码簿登录项中包含许多证件。这些证件有两种类型。第一种是 X 的正向证件，它是由其他发证机构产生的。第二种是由 X 本身产生的反向证件，它们是其他发证机构的合格的公用钥匙。根据这些证件，用户能够逐点建立凭证路径。

7.5 允许某一个用户获得另一个用户的公用钥匙所需要的证件表称为*凭证路径*。表中的每一个项目是下一个项目的发证机构的证件。由 A 至 B 的一条凭证路径（记作 A→B）如下：

— 以 CA（A）为某一实体 X^1 颁发的证件开始，即 CA（A）$\ll X^1 \gg$；

— 继之以更多的证件 $X^i \ll X^{i+1} \gg$；然后

— 以 B 的证件结束。

在 DIT 中，一条凭证路径是在希望进行验证的两个用户之间的多个受信任点构成的一条不间断逻辑链。用户 A 和 B 获得路径 A→B 和 B→A 使用的方法可能有差别。一种方法是组织一个 CA 的分级式体系，这个体系不一定同 DIT 的分级式体系中的一部分或全部吻合。这个方法的优点在于，那些在分级式体系中拥有 CA 的用户可以利用号码簿，在用户与用户之间建立一条凭证路径而无须预先知道任何情报。为此，每一个 CA 可贮存一份证件和一份指定给对应上级 CA 的反向证件。

7.6 证件在号码簿登录项中是做为下列属性类型保存的：**UserCertificate**，**CACertificate** 和 **CrossCertificatePair**。这些属性类型均为号码簿所掌握。对这些属性可利用适用于其他属性的同样协议进行操作。这些属性类型的定义在本建议§3.3中描述；这些属性类型的描述如下：

```
UserCertificate          ::=   ATTRIBUTE
      WITH ATTRIBUTE-SYNTAX Certificate
CACertificate            ::=   ATTRIBUTE
      WITH ATTRIBUTE-SYNTAX Certificate
CrossCertificatePair     ::=   ATTRIBUTE
      WITH ATTRIBUTE-SYNTAX CertificatePair
CertificatePair          ::=
      SEQUENCE{
            forward [o] Certificate OPTIONAL
            reverse [1] Certificate OPTIONAL
            -- at least one must be present --}
```

一个用户可从一个或多个发证机构那里获得一个或多个证件。每一个证件携带发证机构的名称。

7.7 一般情况下，在用户能够彼此相互验证之前，号码簿必须提供完整的凭证和凭证返回路径。但是，在实践上可利用下列方法在特定验证实例中将必须从号码簿里获得的信息加以压缩：

a) 倘若希望验证的两个用户是由同一个发证机构提供服务的，则凭证路径极为简单，用户可直接相互打开对方的证件；

b) 倘若用户的CA是个分级式体系，则用户可将所有发证机构的公用钥匙、证件和反向证件贮存于用户与DIT根之间。典型的例子是，用户只须掌握三至四个发证机构的公用钥匙和证件就足够。从而使用户只需要从公共信任点来获得凭证路径；

c) 倘若一个用户经常同某些用户通信，而后者是由其他特定CA获得证件，则该用户应知道至该CA的凭证路径和从该CA的返回凭证路径，因此本用户只须从号码簿中获得其他用户本身的证件即可；

d) 发证机构可通过双边协商彼此相互证明。其结果是缩短凭证路径；

e) 倘若两个用户以前曾经相互通信并彼此知道对方的证件，则他们能够不求助于号码簿而进行验证。

在任何情况下，当用户根据凭证路径知道对方的证件之后，他们必须检验收到的证件的有效性。

7.8 （例子）图4/X.509表示一个DIT片断的假想实例。其中CA是个分级式体系。假设每个用户除了知道CA显示的信息外，还知道他的发证机构的公用钥匙以及自己的公用钥匙和密钥。

7.8.1 倘若用户的CA是个分级式体系，则A能够从号码簿上获得下列证件和建立至B的凭证路径：

$X\ll W\gg$，$W\ll V\gg$，$V\ll Y\gg$，$Y\ll Z\gg$，$Z\ll B\gg$，

当A获得这些证件之后，他就能够顺序打开凭证路径，给出B的证件内容，包括Bp：

$Bp = Xp \cdot X\ll W\gg W\ll V\gg V\ll Y\gg Y\ll Z\gg Z\ll B\gg$

一般，为了建立自B至A回程的凭证路径，A必须从号码簿里获得下列证件：

$Z\ll Y\gg$，$Y\ll V\gg$，$V\ll W\gg$，$W\ll X\gg$，$X\ll A\gg$。

当 B 从 A 获得这些证件时，他就能够顺序打开回程的凭证路径并给出 A 的证件内容，包括 Ap：
Ap＝Zp・Z≪Y≫Y≪V≫V≪W≫W≪X≫X≪A≫。

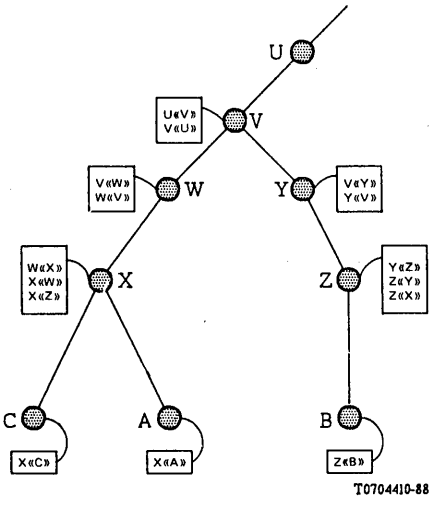


图 4/X.509

CA 分级式体系的假想例

7.8.2 应用§7.7 所述的优化条件：

a) 设以 A 与 C 为例：二者均掌握 X_p，故 A 只须直接获得 C 的证件。打开凭证路径简化为
Cp＝Xp・X≪C≫
且打开回程的凭证路径，简化为：
Ap＝Xp・X≪A≫

b) 假设 A 知道 W≪X≫，Wp，V≪W≫，Vp，U≪V≫，Up，等，则 A 为了建立凭证路径必须将从号码簿获得的信息简化为：
V≪Y≫，Y≪Z≫，Z≪B≫
而 A 为了建立回程的凭证路径，必须从号码簿那里获得的信息简化为：
Z≪Y≫，Y≪V≫

c) 假设 A 经常同某些用户通信，而后者是从 Z 那里获得证件，则 A 能知道（除自上述 b）获知的公用钥匙之外）V≪Y≫，Y≪V≫，Y≪Z≫，和 Z≪Y≫。因此，为了同 B 通信，A 只须从号码簿那里获得 Z≪B≫即可；

d) 假设由 X 与 Z 获得证件的用户经常通信，则在号码簿登录项中将为 X 保存 X≪Z≫，反则亦然（如图 4/X.509 所示）。倘若 A 要验证 B，则 A 只须获得下列证件：
X≪Z≫，Z≪B≫
来建立凭证路径，和 Z≪X≫
来建立回程的凭证路径；

e) 假设用户 A 和 C 以前曾经通信并彼此掌握对方的证件，则他们可直接分别使用对方的公用钥匙，即

Cp＝Xp・X≪C≫和

Ap＝Xp・X≪A≫。

7.8.3 在大多数情况下，发证机构的体系不采用分级式结构。参看图 5/X.509 的假想实例。假设用户 D 从 U 那里获得了证件，希望验证用户 E，而后者是从 W 那里获得了证件，则用户 D 的号码簿登录项将包含证件 U≪D≫，而用户 E 的登录项则将包含证件 W≪E≫。

假设，V 为一个 CA，该 CA 以前曾经同 CA U 和 W 以可信任的方式交换过公用钥匙，则结果是，证件 U≪V≫，V≪U≫，W≪V≫和 V≪W≫被生成并贮存于号码簿里。假设，U≪V≫和 W≪V≫被贮存于 V 的登录项，V≪U≫被贮存于 U 的登录项，V≪W≫被贮存于 W 的登录项。

用户 D 必须寻找至 E 的凭证路径，为此他可采用各种策略：一种策略是把用户和 CA 当做节点，把证件当做有方向性的弧线，在这种情况下，D 必须在图中寻找一条由 U 至 E 的路径，例如 U≪V≫，V≪W≫，W≪E≫。当找到了这条路径之后，就可以建立回程的路径 W≪V≫，V≪U≫，U≪D≫。

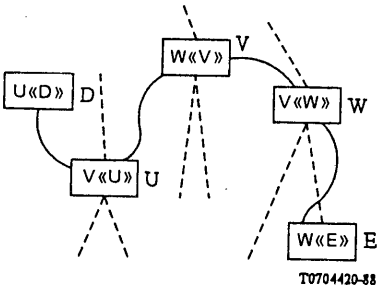


图 5/X.509

非分级式凭证路径的一个例子

8 数字签名

本节不欲规定通用的数字签名标准，而仅规定在号码簿里签署证件的手段。

8.1 签署信息（info）的方法是先将信息的摘要译成密码，然后把它附于该信息。摘要是采用单向散列函数生成的，然后采用签署者的密钥把它译成密码（见图 6/X.509）如下：

X｛Info｝＝Info，Xs［h（Info）］

注 — 采用密钥进行编码是为了保证签名不被伪造。单向散列函数的性能保证不能用假情报冒充真情报来获得相同的散列结果（即相同的签名）。

8.2 接受署名信息的用户使用下列方法进行检验：

— 使用单向散列函数对收到的信息进行检验；

— 把检验结果同使用签署者的公用钥匙进行译码而获得的结果相比较。

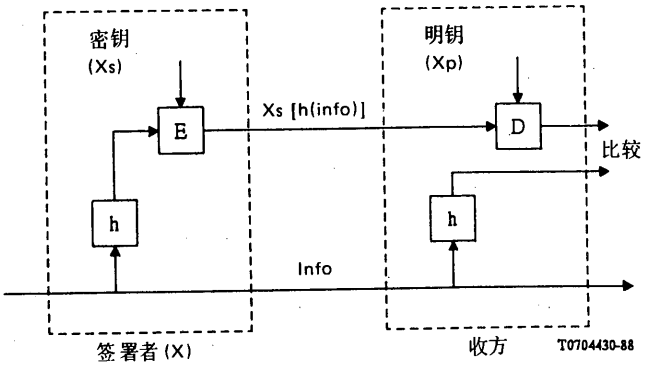


图 6/X.509

数 字 签 名

8.3 本验证框架不硬性规定只使用一种单向散列函数进行签署。本框架应适用于任何合适的散列函数，并支持未来由于密码术、数学技术或计算能力的发展而导致使用方法的变化。但是，准备验证的两个用户必须支持同一个散列函数，以确保验证正确无误。因此，在一组相关应用的范围内选择单一函数有助于扩大能够参与安全验证和通信的用户群。附件 D 给出散列函数的一个范例。

署名信息包含一些指示代码。这些指示代码用于识别那些在计算数字签名中使用的散列函数和密码算法。

8.4 描述某些数据项的编码可使用下列 ASN.1 宏：

```
ENCRYPTED MACRO   ::=
BEGIN
TYPE NOTATION     ::=  type (ToBeEnciphered)
VALUE NOTATION    ::=  value (VALUE BIT STRING)
END
```

为了生成比特串的值，应取构成 **ToBeEnciphered** 类型值的完整编码（使用 ASN.1 基本编码规则）的八位组，使用编码规程对这些八位组进行处理。

注 1 — 编码规程要求使用统一的算法，包括算法中的所有参数。例如所需要的钥匙、初始值以及填充（padding）指示。编码规程的责任包括规定数据接收者和发送者达到同步使用的手段，发送者可能将信息包含在拟发送的比特中。

注 2 — 编码规程要求以一串八位组做为输入，并生成一串比特做为其结果。

注 3 — 关于数据发送者和接收者对安全保护编码算法和它的参数的协商机制不属于本建议的讨论范围。

8.5 在必须把签名附于数据类型的情况下，可使用下列 ASN.1 宏来定义特定数据类型，即由签署一给定数据类型获得的数据类型。

```
SIGNED MACRO        ::=
BEGIN

TYPE NOTATION       ::=   type (ToBeSigned)

VALUE NOTATION      ::=   value (VALUE

      SEQUENCE{
            ToBeSigned,
            AlgorithmIdentifier,
            -- of the algorithm used to compute
            -- the signature
            ENCRYPTED OCTET STRING
            -- where the octet string is the result
            -- of the hashing of the value of
            -- 'ToBeSigned' --}

END -- of SIGNED.          )
```

8.6 在只需要签名的情况下，可使用下列 ASN.1 宏来定义特定数据类型，即由签署一给定数据类型获得的数据类型。

```
SIGNATURE MACRO  ::=
BEGIN

TYPE NOTATION       ::=   type (OfSignature)

VALUE NOTATION      ::=   value (VALUE

      SEQUENCE{
            AlgorithmIdentifier,
            -- of the algorithm used to compute
            -- the signature
            ENCRYPTED OCTET STRING
            -- where the octet string is a function (e.g. a
            -- compressed or hashed version) of the
            -- value 'OfSignature', which may include
            -- the identifier of the algorithm used to
            -- compute the signature --}

END -- of SIGNATURE.     )
```

8.7 为了能够在分布式环境中证实 **SIGNED** 和 **SIGNATURE** 类型，需要采用一种特殊编码。**SIGNED** 或 **SIGNATURE** 数据值的特殊编码应使用建议 X.209 定义的基本编码规则，并受下列条件的限制：

a) 必须使用定界形式的长度编码，以最少的八位组进行编码；

b) 对串类型不得使用结构式编码；

c) 倘若一个类型取的是默认值，则该值可缺席；

d) Set 类型的成分应按其特征值的递增顺序进行编码；

e) Set-of 类型的成分应按其八位组值的递增顺序进行编码；

f) 倘若布尔(Boolean)类型的值为真，则编码时应将其包含的八位组置为 $'FF'_{16}$；

g) 倘若在 Bit String 值的编码中发现最后的一个八位组包含不用的比特，则应将该比特全部置为零；

h) 对实数(Real)类型进行编码时，不须使用基数 8、10 和 16，而且二进制比例数应为零。

9 强验证规程

9.1 概述

9.1.1 验证的基本方法已经在上面阐述，即通过获得一个密钥来证实身份。但是，采用这个方法的验证规程有很多。通常，为了满足特定应用的安全保密政策，应选择适当的规程。本条款描述三种典型的验证规程，有可能用于许多应用场合。

注 — 本建议不提供实现时所需要的详细说明。但另有一些标准将根据特定应用或一般应用的需要提供详细说明。

9.1.2 上述三种验证规程都要求交换验证信息，但交换的次数不同，从而为验证者提供不同程度的安全保障。规程的具体描述如下：

a) 在§9.2中描述的单向验证包括由某用户（A）向另一个用户（B）传送一次信息，并证实：
— A的身份，验证令牌是由A生成的；
— B的身份，验证令牌确实拟送往B；
— 被传送的验证令牌的完整性和“本原性”（即发送次数不超过两次或两次以上的特性）。
同理，倘若在传送验证令牌之后接着还传送其他一些数据，也应证实该数据的完整性和“本原性”。

b) 在§9.3中描述的双向验证包括B向A发送的应答。此外，双向验证还证实：
— 在应答中生成的验证令牌是由B生成的，而且是为了把它发送到A；
— 在应答中发送的验证令牌的完整性和“本原性”；
— （任选）令牌的一部分彼此相互保密；

c) 在§9.4中描述的三向验证还包括由A至B的进一步传送。三向验证同双向验证证实同样的特性，但前者不需要对关联时戳进行检验。

在执行强验证的所有场合，A在进行任何信息交换之前必须先获得B的公用钥匙，以及由B至A回程的验证路径。这有可能包括上面§7所述的号码簿访问。下述规程不重复这种访问的描述。

只有当在本地环境中使用同步时钟时，或根据双边协议需要对时钟保持逻辑同步时，才涉及到下文所述的时戳检验。在任何情况下，建议采用通用协调时间（UTC）。

在下述三种验证规程中，均假设A方业已将验证路径上的所有证件检查完毕。

9.2 单向验证

本验证法包括下述步骤，如图7/X.509所示。

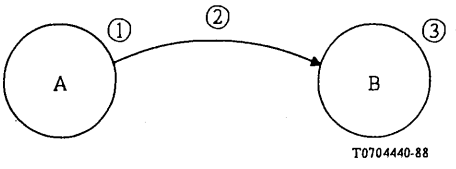


图 7/X.509

单 向 验 证

1） A 生成一个非重复的数目字 r^A，以便用它来检测破坏性重播和防止伪造；

2） A 向 B 发送下列信息：

B→A，A {t^A，r^A，B}

其中，t^A 是一个时戳。t^A 包含一个或两个日期：令牌的生效时间（任选）和截止日期。另一种是，倘若数据源验证“SgnData”拟由数字签名提供，则：

B→A，A {t^A，r^A，B，SgnData}

在准备传递拟用来做密钥的信息的场合（这种信息称为“encData”），应使用下列信息：

B→A，A {t^A，r^A，B，sgnData，Bp [encData]}。

采用“encData”做为密钥意味着必须谨慎选择信息，例如不论采用哪一种密码系统（如在“sgnData”令牌字段中所示），均须采用强验证钥匙。

3） B 完成下列动作：

a） 由 B→A 获得 Ap，检查 A 的证件是否已过期；

b） 检验签名，从而检验署名信息的完整性；

c） 检查 B 是否是期望的接受者；

d） 检查时戳是否是“当前”的时戳；

e） 任选检查 r^A 是否未被重播。为此，令 r^A 包含一个序贯部分，并由本地实施检查它的值是否是唯一的。

在任何情况下，B 方应当以明码保存序贯部分和时戳 t^A，并在时间范围 t^A 的过程中保留令牌的散列部分。

9.3 双向验证

双向验证包括下列步骤，如图 8/X.509 所示。

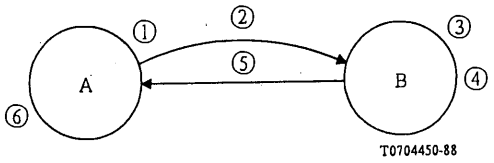


图 8/X.509

双向验证

1) 与§9.2同。

2) 与§9.2同。

3) 与§9.2同。

4) B生成一个不重复的数目字 r^B，其使用目的与 r^A 相似。

5) B向A发送下列验证令牌：

B $\{t^B, r^B, A, r^A\}$

其中，t^B 为一个时戳，其定义与 t^A 相同。

另一种是，倘若数据源验证“SgnData”拟由数字签名提供，则；

B $\{t^B, r^B, A, r^A, \text{sgn Data}\}$

在准备传递拟用来做密钥的信息的场合（这信息称为“encData”），应使用下列信息：

B $\{t^B, r^B, A, r^A, \text{sgnData}, Ap\ [\text{encData}]\}$。

使用“encData”做为密钥意味着必须谨慎选择信息，例如不论采用哪一种密码系统（如在“SgnData”令牌字段中所示），均须采用强验证钥匙。

6) A完成下列动作：

a) 检验签名，从而检验署名信息的完整性；

b) 检查A是否是期望的接受者；

c) 检查 t^B 是否是“当前”的时戳；

d) 任选检查 r^B 是否未被重播［见§9.2中的步骤3）e)］。

9.4 三向验证

三向验证包括下列步骤，如图9/X.509所示。

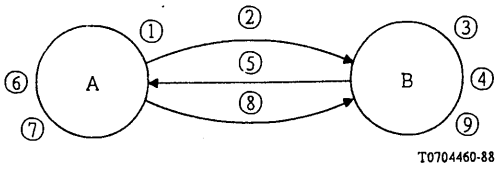


图 9/X.509

三向验证

1） 与§9.3同。
2） 与§9.3同。时戳 t^A 可能为零。
3） 与§9.3同，唯不需要检查时戳。
4） 与§9.3同。
5） 与§9.3同。时戳 t^B 可能为零。
6） 与§9.3同，唯不需要检查时戳。
7） A检查收到的 r^A 是否与发送的 r^A 相同。
8） A向B发送下列验证令牌：
A $\{r^B\}$。
9） B完成下列动作：
a） 检查签名，从而检查署名信息的完整性；
b） 检查收到的 r^B 是否与由B发送的 r^B 相同。

10 钥匙和证件的管理

10.1 钥匙对的生成

10.1.1 具体实施方案中的总安全保密管理政策会定义钥匙对的寿命周期，故此内容不属于本验证框架的范围。但是这里有必要指出：对于总安全保密至关重要的是，所有的密钥必须保密，只有该密钥的所有者才允许知道。

由于钥匙数据不便于人类用户记忆，故必须采用便于贮存和携带的方法。一种可用的机制是采用所谓“Smart Card”。这种卡片可用来装入用户的密钥和（任选的）公用钥匙、用户证件以及发证机构的公用钥匙的拷贝。拟使用这种卡片的用户必须能够报身份证的号码PIN（Personal Identification Number）和出示该卡片，此外必须掌握领取该卡片的方法，这样做的目的是为了提高系统的安全保密度。但是，贮存上述数据的精确方法不属于本建议所涉及的范围。

10.1.2 用户的钥匙对共有三种生成方法，如§10.1.2.1至§10.1.2.3所述。

10.1.2.1 用户生成自己的钥匙对。这种方法的优点是用户的密钥永远不会被透露给其他实体，但是，用户必须掌握一定的技巧，如附件C所述。

10.1.2.2 钥匙对是由第三方生成。第三方必须使用在物理上安全保密的手段向用户颁发密钥，然后主动消毁生成钥匙对所涉及到的全部信息以及该钥匙本身。务必采用适当的安全保密物理措施，保证第三方和数据操作免受干扰。

10.1.2.3 钥匙对是由CA生成的。这是§10.1.2.2的特例，而且上述考虑同样适用于这种生成方法。

注 一 对用户而言，发证机构已经显示出可信赖的功能，并将接受必要的安全保密物理措施。本方法的优点在于不要求为了验证而必须把数据安全地送到CA。

10.1.2.4 使用的密码系统对钥匙的生成提出特定的（技术）限制条件。

10.2 证件的管理

10.2.1 一个证件把公用钥匙同说明用户的唯一可识别名联系起来。因此：
a） 发证机构在建立用户证件之前必须对用户的身份证感到满意；

b) 发证机构不得使用同样的名称为两个用户颁发证件。

10.2.2 证件是以离线方式生成的而不须使用自动询问/响应机制。这种证件生成法的优点在于发证机构CA的密钥是永远不透露的，除非是在隔离情况下和在物理上安全的CA内部，这样，除非通过对CA本身的袭击，是无法查到CA的密钥的，因此不可能泄密。

10.2.3 保证送往发证机构的信息不被泄露是至关重要的，故务必采取适当的物理安全保密措施。在这方面：

a) 倘若CA使用被盗用过的公用钥匙给用户颁发证件，则这是对安全保密规定的严重违犯；

b) 倘若采用§10.1.2.3的手段生成钥匙对，则不需要安全保密传送；

c) 倘若采用§10.1.2.1或§10.1.2.2的手段生成钥匙对，则用户可使用不同的方法(在线或离线方式)以安全保密方式将其公用钥匙传递给CA。在线方式为由用户与CA之间完成的远程操作增加了灵活性。

10.2.4 证件是公开提供的一则信息，把证件送往号码簿不需要使用特殊的安全保密措施。由于证件是离线发证机构为将收到证件拷贝的用户提供的，用户只须在访问号码簿时将此信息贮存于其号码簿登录项中。另一种选择是：CA可为用户存放证件，在此情况下该代理必须得到适当的访问权。

10.2.5 证件是有寿命的、当寿命到期之后证件就过期。为了提供服务的连续性，CA应保证及时用新证件更换已经过期或即将过期的证件。有关的诸方面如§10.2.5.1和§10.2.5.2所述。

10.2.5.1 证件的有效期是可以设计的，使得在旧的证件过期之时新证件开始生效，或者允许新、旧证件重叠。后一种方法可使CA避免必须在同一个失效日期内装配和颁发大量的证件，致使这些证件可能不够用。

10.2.5.2 过期的证件通常需要从号码簿里删除。若提供非拒绝性数据业务，安全保密政策和CA负责把旧证件保留一段时期。

10.2.6 在证件过期之前照例可以废除该证件。例如，假设用户的密钥已被泄漏，或该用户已不复为CA所证实，或者假设CA的证件已被泄漏。有关的诸方面，如§10.2.6.1至§10.2.6.4所述。

10.2.6.1 有关用户证件或CA证件的废除应由CA公布，并在适当场合提供新证件。然后由CA通过某种离线规程向证件的所有者报告证件的废除情况。

10.2.6.2 CA应保存下列表格：

a) CA颁发的证件之中已被废除的证件一览表和各证件上盖的时戳；

b) CA所掌握和证实的全部被废除的证件一览表和各证件上盖的时戳。

既使是空的，这两种表也应当存在。

10.2.6.3 号码簿因受CA证件废除表的影响所需要的维护工作应由号码簿及其用户根据安全保密政策负责完成。例如，用户可通过以旧换新的方法更换证件，来修改其客体登录项。然后利用新证件去验证号码簿用户。

10.2.6.4 证件废除表（“黑名单”）做为“**CertificateRevocationList**”和“**AuthorityRevocationlist**”属性类型，保存在登录项内。这些属性的操作与其他属性相同。这些属性类型的定义如下：

```
CertificateRevocationList    ::=  ATTRIBUTE
     WITH ATTRIBUTE-SYNTAX CertificateList
AuthorityRevocationList      ::=  ATTRIBUTE
     WITH ATTRIBUTE-SYNTAX CertificateList
CertificateList              ::=  SIGNED SEQUENCE{
     signature AlgorithmIdentifier,
     issuer Name,
     lastUpdate UTCTime,
     revokedCertificates
          SIGNED SEQUENCE OF SEQUENCE{
               signature AlgorithmIdentifier,
               issuer Name, CertificateSerialNumber subject,
               revocationDate UTCTime}
                         OPTIONAL}
```

注 1 — 整个证件表的检验属于本地决策。

注 2 — 倘若非拒绝性数据业务取决于由 CA 提供的钥匙，则该业务必须保证 CA 的所有相关钥匙（被废除或过期的）和盖有时戳的废除表是由当前权威机构负责保存和证实的。

附 件 A

（附于建议 X.509）

安全保密要求

本附件不是本建议的组成部分。

[与本题有关的资料有 OSI 7498 - Information Processing Systems-OSI Reference Model-Part2，Security Architecture]

许多 OSI 应用、CCITT 定义的服务和非 CCITT 定义的服务都有可能提出安全保密要求。这样的要求是为保护信息传送不受各种可能的威胁而提出的。

A.1 威胁

一些常见的威胁如下：

a) 身份证窃获：发现有一个或多个参与通信的用户的身份证已经被别人非法使用；

b) 冒充：某一个用户，为了获取信息或获得额外权力冒充别的用户；

c) 重播：把信息录下来准备过一段时间重播；

d) 数据窃获：在通信过程中，无权用户偷看用户数据；

e) 非法处理用户数据：在通信过程中，无权用户擅自更换、插入、删除用户数据或扰乱用户数据的排列顺序；

f) 否认：用户否认参与过某次通信的一部分或全部过程；

g) 拒绝服务：妨碍或中断通信，或延误了对时间性要求较高的操作；

注 — 这种对安全的威胁比较一般，是由个别应用或非法中断的意图决定的，因此不能明显地列入验证框架的范围。

h) 错误的路由选择：为一个用户与另一个用户之间的通信选择不合适的路径；

注 — 错误的路由选择自然发生在开放系统互连第 1 至 3 层。因此错误的路由选择不属于验证框架的范围。但是，通过使用验证框架所提供的适当的安全服务，有可能避免错误路由选择引起的后果。

i) 业务量分析：对用户之间的通信的有关信息进行检测和观察（例如，有/没有，频率，方向，序列，类型，数量，等）。

注 — 业务量分析的威胁自然不限于某一个 OSI 层。因此，验证框架通常不包括业务量分析。但是，通过生成附加的不可懂业务量（业务量填充）、使用加密数据或随机数据，业务量分析是可加以部分保护的。

A.2 安全保密服务

为了防止已觉察到的威胁，有必要提供各种安全保密服务。验证框架提供的安全保密服务是利用本附件 A.3 所述的安全机制实现的。

a) 对等实体的验证：本服务是用于确证某一个用户在某一通信瞬间正是所要寻找的那个用户。可以申请两种不同的对等实体验证服务：

— 单实体验证（数据源实体验证或数据宿实体验证）；

— 双边验证：互相通信的两个用户彼此互相验证。

当申请对等实体验证服务时，两个用户应当协商是否要求保护自己的身份证。

验证框架支持对等实体验证服务。对等实体验证服务可用来保护用户身份证不被别人冒充和重播；

b) 访问控制：这种服务可用来保护资源不被无权用户滥用。访问控制服务是由号码簿或其他应用提供的，因此不属于验证框架所关心的范围；

c) 数据保密：这种服务可用来保护数据不被非法泄漏。验证框架支持数据保密服务。它可用来防止数据被窃获；

d) 数据完整性：这种服务保证数据在通信中保持完整。验证框架支持数据完整性服务。它可用来检测和保护数据不受非法处理；

e) 不可否认：这种服务证明数据的完整性和数据源的可靠性，二者的关系是无法伪造的，而且随时都可由第三方证实。

A.3 安全保密机制

本节介绍的安全保密机制提供 A.2 所述的安全保密服务。

a) 交换验证：验证框架分为以下两个等级：

— 简单验证：根据发起者提供的名称和口令，由收方进行验证；

— 强验证：采用密码技术来保护验证信息的交换。在验证框架中，强验证是按不对称方案描述的。

验证交换机制是用来支持对等实体验证服务的；

b) 加密：验证框架要求在数据传送过程中提供数据加密。可采用对称或不对称方案加密。这两种加密方案所需要的钥匙交换可在早先进行的验证交换中完成，也可以在进行通信之前的任何时候用离线方式完成。后一种情况不属于验证框架的范围。加密机制支持数据保密服务；

c) 数据完整性：这个机制包括对准备传送的数据的压缩字符串进行加密。然后把这个信息包连同普通（未加密）数据送到收方。收方对该普通数据重复进行压缩和加密，然后把结果同发起者生成的数据相比较证明其完整性。

数据完整性机制可通过对被压缩的普通数据以对称或不对称方式加密的办法来提供（使用对称方式时，数据的压缩和加密可同时进行）。验证框架不使用明显方式提供这个机制。但是，当使用不对称方式提供数字签名机制的一部分时仍然充分提供数据完整性机制。

数据完整性机制支持数据完整性服务。它部分地支持防否认服务（该服务同样需要数字签名机制以保证它的要求全部得到满足）；

d) 数字签名：这种机制包括使用发起者的密钥对准备传送的数据的压缩字符串进行加密。数字签名和普通数据一起被送到收方。在类似数据完整性机制的情况下，这个信息由收方处理，来检验它的完整性。数字签名机制同样证明发起者的真实性和发起者与被发送的数据之间的明确关系。

验证框架使用不对称方式支持数字签名机制。

数字签名机制支持数据完整性服务和防否认服务。

A.4 使用安全保密服务以防威胁

本附件后面的表列出每一个安全保密服务所能防止的一些有害威胁。表中的星号表示某一种安全保密服务能够保护数据免受某种威胁。

A.5 机制和安全保密服务的协商

为了在某一个通信实例中提供安全保密措施，对所要求的安全保密服务的内容应进行协商。这就需要对提供上述安全服务所需要的安全机制的类型和安全参数进行协商。机制和参数协商所需要的规程可做为一般的连接建立规程的组成部分，或做为独立进程来描述。有关这些协商规程的详细说明不在本附件中描述。

服　务

威胁	实体验证	数据保密	数据完整性	防否认
窃获身份证	* （若申请）			
数据窃获		*		
冒充	*			
重播	* （身份证）		* （数据）	*
处理			*	*
否认				*

附　件　B

（附于建议 X.509）

公用钥匙加密技术简介

本附件不是本建议的一个组成部分。

在传统的密码系统中，一般来说，由发起者把秘密情报译成密码使用的钥匙同合法的收方对该情报进行解码使用的钥匙是一样的。

但是，在公用钥匙密码系统（PKCS）中提供的是一对钥匙：一把钥匙专门用于把数据译成密码，另一把专门用于解密码。每一对钥匙同一特定用户 X 有关联。其中的一把钥匙叫做公用钥匙（Xp），是对公众公开的，任何用户都可以用它把数据译成密码。但是，只有掌握互补密钥（Xs）的用户 X 才有可能解该数据的密码（这可用符号表示：D=Xs [Xp [D]]）。要想根据从公用钥匙所获得的情报来导引密钥，从算法上是行不通的。因此，任何用户都能够利用 Xp 把要发送的一段情报译成密码，以保证只有用户 X 能够找到这段情报。通过扩展，两个用户能够彼此利用对方的公用钥匙，把要发送的数据译成密码，以便进行秘密通

信，如图 B-1/X.509 所示。

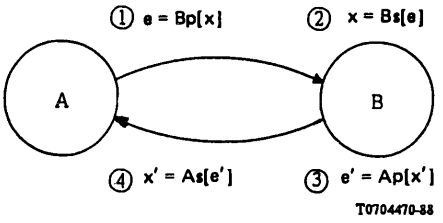


图 B-1/X.509

利用 PKCS 交换秘密情报

用户 A 拥有公用钥匙 Ap 和密钥 As，用户 B 也拥有另一对钥匙 Bp 和 Bs。用户 A 和 B 彼此知道对方的公用钥匙，但互不知道对方的密钥。因此，A 和 B 通过下列步骤交换秘密钥匙（参见图 B-1/X.509）：

1） A 希望把秘密情报 x 送到 B。为此，A 利用 B 的加密钥匙把 x 译成密码并把译成密码的情报 e 发送到 B。用符号表示如下：

$e=Bp\ [x]$

2） B 现在可以利用解密密钥 Bs 对上述加密情报 e 进行解码，以获得情报 x。请注意，用户 B 是 Bs 的唯一所有者，因此由于这把密钥永远不会公开或发送出去，其他任何用户是不会获得该份情报 x 的。Bs 的所有权确定 B 的身份。解码操作可如下表示：

$x=Bs\ [e]$，或 $x=Bs\ [Bp\ [x]]$

3） 同理，现在 B 可以利用 A 的加密钥匙 Ap 向 A 发送秘密情报 x′：

$e'=Ap\ [x']$。

4） A 通过对 e′ 进行解码而获得 x′：

$x'=As\ [e']$，或 $x'=As\ [Ap\ [x']]$。

通过上述办法，A 和 B 进行了秘密情报 x 和 x′ 的交换。只要 A 和 B 的密钥不公开，任何人除 A 和 B 之外是永远不会获得上述情报的。

这样的情报交换以及在两个用户之间传送秘密情报的操作，可用来验证对方的身份。例如，A 和 B 的身份可分别通过它们的解密专用密钥 As 和 Bs 来识别。A 可以根据由 B 回送的信息 x′ 中的一部分 x 情报确定 B 是否拥有解密专用密钥 Bs。这就向 A 表明通信是和 Bs 的拥有者进行的。同理，B 可通过类似的办法检验 A 的身份。

某些 PKCS 具备这样的特性，即加密和解密的步骤可以颠倒，如在 $D=Xp\ [Xs\ [D]]$ 中。这使得原来只有用户 X 才能发送的一份情报，任何用户（拥有 Xp 的用户）都能读取。这就有可能用它来检验情报的来源，和做为数据签名的基础。只有具备这种（可转换性）特性的 PKCS 才适用于这种验证框架。计算的方法在附件 C 中描述。

更详细的资料请参阅：

DIFFIE, W. and HELLMAN, M. E. (November 1976) -New Directions in Cryptography, *IEEE Transactions on Information Theory*, IT-22, No. 6.

附 件 C

（附于建议 X.509）

RSA 公用钥匙密码系统

本附件不是本建议的组成部分。

注 — 本附件描述的密码系统是众所周知的“RSA”系统，它是由 R. L. Rivest，A. Shamir 和 L. Adleman 发明的。

C.1 应用范围

本文不准备详细讨论 RSA。但是，本文将简单介绍有关的方法，它是采用取幂模算法导出的。

C.2 参考文献

更详细的资料请参阅下列文献：

1） 概述

RIVEST，R・L・，SHAMIR，A. and ADLEMAN，L.（February 1978）-A Method for Obtaining Digital Signatures and Public-Key Cryptosystems，*Communications of the ACM*，21，2，120-126

2） 钥匙生成参考资料

GORDON，J.-Strong RSA Keys，*Electronics Letters*，20，5，514-516

3） 解码参考资料

QUISQUATER，J. J. and COUVREUR，C.（October 14，1982）-Fast Decipherment Algorithm for RSA Public-key Cryptosystems，*Electronics Letters*，18，21，905-907

C.3 定义

a） *public key* 公用钥匙：一对参数，它是由公用指数和算术模组成的；

注 — 定义为 **BIT STRING**（见附件 G）的 ASN.1 数据要素 **subjectPublicKey**，在 RSA 的情况下应理解为下列类型：

SEQUENCE {INTEGER，INTEGER}

其中，第一个整数是算术模，第二个整数是公用指数。该序列是用 ASN.1 基本编码规则表示的。

b） *secret key* 密钥：一对参数，它是由秘密指数和算术模组成的。

C.4 符号与缩写词

X，Y 数据块，它在算术上小于模数

n 算术模，

e 公用指数，

d 秘密指数，

p，q 素数，其乘积构成算术模（n），

注 — 虽然优先选择的素数是两个，但不排除使用由三个或多个素数构成的模。

mod n 算术模 n。

C.5 说明

不对称算法使用幂函数转换数据块，使得：

$Y=X^e \bmod n \qquad 0\leqslant X<n$

$X=Y^d \bmod n \qquad 0\leqslant Y<n$

上式被下列条件所满足：

$ed \bmod \mathrm{lcm}\ (p-1,\ q-1)=1$，

$ed \bmod (p-1)(q-1)=1$。

为了执行本进程，应当把一个数据块理解为一个整数。为此，可把整个数据块当做一个有顺序的比特序列（例如其长度为l）。然后令整数等于上述比特之和，其中第一个比特的加权为 2^{l-1}，后随的每一个比特的加权除以 2（最后一个比特的加权为 1）。

数据块的长度应当等于八位组（octet）的最大数目，它包含的比特的总数少于模数。不完整的数据块应使用适当的方法填充。作为填充，数据块的数量可任意增加。

C.6 安全保密要求

C.6.1 钥匙的长度

众所周知，钥匙的适当长度是有可能随时间而变化的，这取决于硬件的成本和提供的情况、消耗的时间、技术进步和所需要的安全保密程度。建议在初期阶段取 512 比特做为长度值 n，关于这个值有待进一步研究。

C.6.2 钥匙的生成

RSA 的安全保密性能取决于对 n 因子的分解难度。有许多种算法可用于这种操作，为了防止使用目前已知的任何技术，务必根据下列准则［见参考文献 2)，C.2］精心选择 p 和 q 的值：

a) 必须随机选择上述值；

b) 必须选择大的数值；

c) 该值必须是素数；

d) $|p-q|$必须很大；

e) $(p+1)$必须具有很大的素数因子；

f) $(q+1)$必须具有很大的素数因子；

g) $(p-1)$必须具有很大的素数因子，如 r；

h) $(q-1)$必须具有很大的素数因子，如 s；

i) $(r-1)$必须具有很大的素数因子；

j) $(s-1)$必须具有很大的素数因子。

生成包含 d、e 和 n 的公用钥匙和密钥，如本建议§3.3 和§4.1 定义的“Xp”和“Xs”之后，照例应当把 p 和 q 值以及产生的其他数据，如 $(p-1)(q-1)$ 乘积和很大的素数因子全部消毁。但是，在本地保存 p 和 q 可进一步提高解译量二至四倍。是否要保存 p 和 q 属于本地决策［参考文献 3)］。

为了防止对数据的侵犯，即取出 e 次根 mod n 来暴露秘密文件，必须保证 $e>\log_2(n)$。

C.7 公用指数

为了尽量缩短实际要分配的那部分公用钥匙的长度，以便减少传输容量和转换的复杂性（见注 1），公

用指数（e）在所有的环境里可以公用。

指数 e 应当足够大，但是，就取幂时的处理时间和存储容量而言，应考虑提高效率。倘若希望采用一个固定的公用指数 e，则采用 Fermat 数 F_4 会带来好处（见注 2）。

$F_4 = 2^{2^4} + 1$

$= 65537$ 十进制，和

$= 1\ 0000\ 0000\ 0000\ 0001$ 二进制

注 1 — 虽然模 n 和指数 e 是公用的，但是该模数不应当为一组用户所公用。知道模“m”、公用指数“e”和秘密指数“d”就足够确定“m”的因子分解。因此，倘若该模数是公用的，则每一个人都能够计算它的因子，从而可以找到任何人的秘密指数。

注 2 — 固定指数应当是很大的素数，但也应当考虑处理效率。Fermat 数 F_4 满足这些条件，例如验证只需 17 个乘法运算，而且比解码速度平均高 30 倍。

C.8 一致性

本附件描述计算公用和秘密函数的算法，但不定义实际运算的方法；因此有可能生产符合本附件的各种不同的产品，而且彼此兼容。

附 件 D

（附于建议 X.509）

散 列 函 数

本附件不是本建议的组成部分。

D.1 散列函数的要求

若希望获得高可靠性和单向性散列函数，必须防止利用输入信息的各种组合就能轻而易举地获得相同的散列（hash）结果。

一个很强的散列函数必须满足下列要求：

a） 该散列函数必须是单向的，即给出一个可能的散列结果，保证在计算上无法建立一个能通过散列算法而获得该散列结果的输入信息。

b） 该散列函数必须具备防碰撞性能，即必须保证在计算上不可能建立能通过散列算法获得同一个结果的两个不同的输入信息。

D.2 散列函数说明

下面的散列函数（“平方-mod n”）完成按块的数据压缩。

计算散列函数的三个步骤如下：

1） 先把准备用散列法进行计算的数据串分成等长的数据块 B。这个长度是根据签署用的不对称密码

系统的特性确定的。当使用 RSA 密码系统时，这个长度（单位为八位组）是最大的整数 1，使得 modulus（模）n，16 $1<\log_2 n$。

2） 为了保证不可逆性，每个数据块的八位组应当一分为二。每一个半截应冠以（“填充”）二进制码。通过这样的分区就可以获得钢性（stiffness）或冗余度，这有助于大幅度增强散列函数的不可逆性特性。由步骤 1 生成的每个数据块沿着模 n 的长度展开。

3） 由步骤 2 获得的每个数据块加到前一个数据块模 2 加、开平方、缩减模 n，直到全部 m 块都处理完毕为止。

其结果为 H_m 值，

其中，$H_0=0$

$H_i=(H_{i-1}\oplus B_i)^2 \bmod n, \qquad 1\leqslant i\leqslant m$

倘若准备用散列法运算的最后一个数据块不完整，可用“1”填充。

附 件 E

（附于建议 X.509）

利用强验证法保护数据不受侵犯

本附件不是本建议的组成部分。

本建议描述的强验证法，为使数据不受如附件 A 所述的威胁提供保护。

另外，强验证法本身也受到如下一系列潜在的威胁：

泄漏用户密钥：强验证的基本原则之一是保护用户密钥的安全。有许多安全保存密钥的实用方法可共用户使用。泄漏密钥的后果仅是有该用户参加的通信遭受破坏。

泄漏 CA 的密钥：强验证的基本原则也包括保护 CA 的密钥的安全。物理安全保密和“需要知道”的方法在这里都适用。泄漏 CA 的密钥造成的后果是：凡是有该 CA 证实的任何用户参加的通信都将受到破坏。

诱导 CA 发放无效证件：各 CA 是采用离线方式操作的，因此可加以保护。CA 的责任是：在建立证件之前应当先检查声称的强凭证是否确实有效。当这道手续出现差错时，其后果是：凡是有持这样的证件的用户和任何接触该无效证件的人参加的通信都将受到影响。

无赖 CA 和用户勾结在一起：这样的串通性破坏行为会使本验证法遭受失败。同时，这将破坏 CA 的信誉。其后果是：有持该 CA 颁发的证件的任何用户参加的通信将受到破坏。

伪造证件：强验证法采用由 CA 签署证件的措施保护证件不被伪造。这种验证法是依靠对 CA 的密钥采取保密性措施实现的。

伪造许可证：强验证法保护许可证不被伪造。其方法是要求发送方在该许可证上签名。这个方法是依靠对发送方的密钥采取保密措施实现的。

许可证的重播：单向和双向验证法保护许可证不被重播，其方法是在许可证上加盖时戳。三向法是通过检查随机数实现的。

对密码系统的冲击：应当考虑到，依靠先进的数字计算理论对密码系统进行有效的密码分析的可能性，导致了增加密钥长度的必要性。

附 件 F

（附于建议 X.509）

数 据 机 密 性

本附件不是本建议的组成部分。

F.1 引言

对数据的机密性处理可以在交换加密专用钥匙之后开始进行。这可以通过§9所述的验证交换或其他的钥匙交换进程来提供，后者不属于本文讨论范围。

数据机密性可采用对称或不对称方式提供。

F.2 采用不对称加密方式的数据机密性

在这种方式下，数据机密性是这样处理的：发起者使用被选收方的公用钥匙把准备发送的数据译成密码，然后由收方使用密钥解密。

F.3 采用对称加密方式的数据机密性

在这种方式下，数据机密性是通过对称加密算法获得的。它的选择不属于验证框架的范围。

当双方完成§9所述的验证交换之后就可以获得用于对称算法的一把钥匙。密钥的选择取决于欲采用的转换。双方必须保证选用的是强验证钥匙。本建议不规定选择的方法。该选择方法显然要经过双方协商或在其他标准中描述。

附 件 G

（附于建议 X.509）

使用 ASN.1 描述的验证框架

本附件是本建议的一部分。

本附件包括在本建议出现的所有 ASN.1 类型、宏和值的定义，并以 ASN.1 模块“**Authentication-Framework**”的形式提供。

```
AuthenticationFramework {joint-iso-ccitt ds(5) modules(1)
                         authenticationFramework(7)}

DEFINITIONS ::=
BEGIN

EXPORTS     AlgorithmIdentifier, AuthorityRevocationList, CACertificate, Certificate,
            Certificates, CertificationPath, CertificateRevocationList, UserCertificate,
            CrossCertificatePair, UserPassword, ALGORITHM,
            ENCRYPTED, PROTECTED, SIGNATURE, SIGNED;
```

```
IMPORTS
    informationFramework, selectedAttributeTypes, upperBounds
        FROM UsefulDefinitions {joint-iso-ccitt ds(5)modules(1)
                        usefulDefinitions(0)}
    Name, ATTRIBUTE,ATTRIBUTE-SYNTAX
        FROM InformationFramework informationFramework

ub-user-passwordFROM Upper Bounds upperBounds;

-- types

Certificate      ::=  SIGNED SEQUENCE{
                        version                  [0] Version DEFAULT 1988,
                        serialNumber             SerialNumber,
                        signature                AlgorithmIdentifier,
                        issuer                   Name,
                        validity                 Validity,
                        subject                  Name,
                        subjectPublicKeyInfo     SubjectPublicKeyInfo}

Version          ::=  INTEGER { 1988(0)}
SerialNumber     ::=  INTEGER

Validity         ::=  SEQUENCE{
                        notBefore.               UTCTime
                        notAfter                 UTCTime}

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

AlgorithmIdentifier   ::= SEQUENCE{
                            algorithm            OBJECT IDENTIFIER,
                            parameters           ANY DEFINED BY algorithm OPTIONAL}

Certificates     ::=  SEQUENCE{
                        certificate              Certificate,
                        certificationPath        ForwardCertificationPath OPTIONAL}

ForwardCertificationPath  ::=   SEQUENCE OF CrossCertificates

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

CertificateList    ::=  SIGNED SEQUENCE{
                            signature            AlgorithmIdentifier,
                            issuer               Name,
                            lastUpdate           UTCTime,
                            revokedCertificates    SIGNEDSEQUENCE OF SEQUENCE{
                                                     signature        AlgorithmIdentifier,
                                                     issuer           Name,
                                                     userCertificate  SerialNumber,
                                                     revocationDate   UTCTime}
                                                              OPTIONAL}
CertificatePair    ::= SEQUENCE{
                          forward [0]            Certificate OPTIONAL,
                          reverse [1]            Certificate OPTIONAL
                          -- at least one of the pair must be present --}

--attribute types

UserCertificate          ::=   ATTRIBUTE
                                 WITH ATTRIBUTE-SYNTAXCertificate

CACertificate            ::=   ATTRIBUTE
                                 WITH ATTRIBUTE-SYNTAXCertificate
```

```
CrossCertificatePair          ::=  ATTRIBUTE
                                      WITH ATTRIBUTE-SYNTAXCertificatePair

CertificateRevocationList     ::=  ATTRIBUTE
                                      WITH ATTRIBUTE-SYNTAXCertificateList

AuthorityRevocationList       ::=  ATTRIBUTE
                                      WITH ATTRIBUTE-SYNTAXCertificateList

UserPassword                  ::=  ATTRIBUTE
                                      WITH ATTRIBUTE-SYNTAX
                                      OCTETSTRING(SIZE(0...ub-user-password))
                                      MATCHES FOR EQUALITY

-- macros

ALGORITHM MACRO               ::=
BEGIN
TYPE NOTATION                 ::=  "PARAMETER" type
VALUE NOTATION                ::=  value(VALUE OBJECT IDENTIFIER)
END -- of ALGORITHM

ENCRYPTED MACRO               ::=
BEGIN

TYPE NOTATION                 ::=  type (ToBeEnciphered)
VALUENOTATION                 ::=  value (VALUE BIT STRING
    -- the value of the bit string is generated by
    -- taking the octets which form the complete
    -- encoding (using the ASN.1 Basic Encoding Rules)
    -- of the value of the ToBeEnciphered type and
    -- applying an encipherment procedure to those octets--

END

SIGNED MACRO                  ::=
BEGIN
TYPE NOTATION                 ::=  type (ToBeSigned)
VALUE NOTATION                ::=  value(VALUE

SEQUENCE{
    ToBeSigned,
    AlgorithmIdentifier, -- of the algorithm used to generate the signature
    ENCRYPTED OCTET STRING
    -- where the octet string is the result
    -- of the hashing of the value of
    -- "ToBeSigned" --}
                    )
END -- of SIGNED

SIGNATURE MACRO               ::=
BEGIN

TYPE NOTATION                 ::=  type (OfSignature)

VALUE NOTATION                ::=  value(VALUE

    SEQUENCE{
        AlgorithmIdentifier,
        -- of the algorithm used to compute the signature
        ENCRYPTED OCTET STRING
        -- where the octet string is a function (e.g. a compressed or hashed version)
        -- of the value "OfSignature", which may include the identifier of the
        -- algorithm used to compute the signature --}
                                                    )
END -- of SIGNATURE

PROTECTED MACRO               ::= SIGNATURE

END -- of Authentication Framework Definitions
```

附 件 H

（附于建议 X.509）

算法客体标识符的参考定义

本附件不是本建议的组成部分。

本附件定义在无正规寄存器的情况下给验证和加密算法分配的客体标识符。一旦有了正规的寄存器，就应当使用它。本定义是以 ASN.1 模块 **AlgorithmObjectIdentifiers** 的形式提供的。

```
AlgorithmObjectIdentifiers   {joint-iso-ccitt ds(5) modules(1)
                              algorithmObjectIdentifiers(8)}
DEFINITIONS ::=
BEGIN

EXPORTS
      encryptionAlgorithm, hashAlgorithm, signatureAlgorithm,
      rsa,squareMod-n,sqMod-nWithRSA;

IMPORTS
      algorithm,authenticationFramework
            FROM UsefulDefinitions{joint-iso-ccitt ds(5)modules(1)
                                   usefulDefinitions(0)}

      ALGORITHM FROM AuthenticationFramework authenticationFramework;

-- categories of object identifier

encryptionAlgorithm OBJECT IDENTIFIER    ::=    {algorithm 1}

hashAlgorithm OBJECT IDENTIFIER          ::=    {algorithm 2}

signatureAlgorithm OBJECT IDENTIFIER     ::=    {algorithm 3}

-- algorithms

rsa    ALGORITHM
       PARAMETER KeySize
       ::= {encryptionAlgorithm 1}

KeySize ::= INTEGER

sqMod-n ALGORITHM
       PARAMETER BlockSize
       ::= {hashAlgorithm 1}

BlockSize ::= INTEGER
sqMod-nWithRSA ALGORITHM
       PARAMETER KeyAndBlockSize
       ::= {signatureAlgorithm 1}

KeyAndBlockSize ::= INTEGER

END -- of Algorithm Object Identifier Definitions
```

号码簿抽象服务定义[1]

(1988 年订于墨尔本)

目　录

0　绪论 …… 83

1　应用范围 …… 83

第 1 部分 — 概述 …… 83

2　参考文献 …… 83

3　定义 …… 83

4　缩写词 …… 85

5　规约 …… 85

第 2 部分 — 抽象服务 …… 85

6　号码簿服务概述 …… 85

7　信息类型 …… 86

8　耦合与释放操作 …… 93

9　号码簿读操作 …… 95

10　号码簿搜索操作 …… 97

11　号码簿修改操作 …… 100

12　差错 …… 103

附件 *A* — ASN.1 中的抽象服务 …… 107

附件 *B* — 号码簿客体标识符 …… 115

---

① 本建议 X.511 与 ISO 9594-3 标准，信息处理系统-开放系统互连-号码簿抽象服务定义是在紧密合作下制定的，而且在技术上是协调的。

0 绪论

0.1 本文和本系列其他文件的目的在于简化信息处理系统之间的互连，以便提供号码簿服务。全套系统，包括它们掌握的号码簿信息，可视为一个整体，称为号码簿。号码簿包含的全部信息叫称为码簿信息库(DIB)，其典型用途是简化客体之间以及同这些客体，或围绕这些客体（如应用实体、人员、终端和分布表）的通信联系。

0.2 号码簿在开放系统互连中扮演重要角色；开放系统的目的是除使用互连标准之外设法只通过很少的技术协议就能实现下列信息处理系统之间的互连：

— 不同厂家；
— 不同管理部门；
— 不同复杂程度；
— 不同年代。

0.3 本建议定义号码簿向用户提供的功能。

0.4 附件 A 提供有关抽象服务的全部定义的 ASN.1 模块。

1 应用范围

1.1 本建议为号码簿定义从外部看得见的抽象服务。

1.2 本建议不提供个别实现或产品的说明。

第 1 部分 — 概述

2 参考文献

建议 X.200 — 开放系统互连基本参考模型
建议 X.208 — 开放系统互连抽象语法符号第 1 集（ASN.1）的规范（ASN.1）
建议 X.500 — 号码簿概念、模型和服务概述
建议 X.501 — 号码簿模型
建议 X.518 — 号码簿分布式操作规程
建议 X.519 — 号码簿协议描述
建议 X.520 — 号码簿属性类型集
建议 X.521 — 号码簿客体类别集
建议 X.509 — 号码簿验证框架
建议 X.219 — 远程操作模型、符号和服务定义
建议 X.229 — 远程操作协议描述
建议 X.407 — 抽象服务定义规约

3 定义

3.1 基本号码簿的定义

本建议使用下列由建议 X.500 定义的术语：

a) *Directory* 号码簿；

b) *Directory Information Base (DIB)* 号码簿信息库；

c) *(Directory) User* （号码簿）用户。

3.2 号码簿模型定义

本建议使用下列由建议 X.501 定义的术语：

a) *Directory System Agent* 号码簿系统代理；

b) *Directory User Agent* 号码簿用户代理。

3.3 号码簿信息库定义

本建议使用下列由建议 X.501 定义的术语：

a) *alias entry* 别名登录项；

b) *Directory Information Tree* 号码簿信息树；

c) *(Directory) entry* （号码簿）登录项；

d) *immediate superior* 直接上级；

e) *immediately superior entry/object* 直接上级登录项/客体；

f) *object* 客体；

g) *object class* 客体类别；

h) *object entry* 客体登录项；

i) *subordinate* 下级；

j) *superior* 上级；

3.4 号码簿登录项定义

本建议使用下列由建议 X.501 定义的术语：

a) *attribute* 属性；

b) *attribute type* 属性类型；

c) *attribute value* 属性值；

d) *attribute value assertion* 属性值判断语。

3.5 名称定义

本建议使用下列由建议 X.501 定义的术语：

a) *alias, alias name* 别名，别名名称；

b) *distinguished name* 可识别名；

c) *(directory) name* （号码簿）名；

d) *purported name* 声称名；

e) *relative distinguished name* 相对可识别名。

3.6 分布式操作定义

本建议使用下列由建议 X.518 定义的术语：

a) *chaining* 连锁操作；

b) *referral* 转移参考。

3.7 抽象服务定义

本建议定义下列术语：

a) *filter* 过滤器：关于某一个登录项的某一个属性的存在或值的判断语，以便缩小搜索范围；

b) *service controls* 服务控制：限定操作性能各个方面的参数，做为抽象操作的一部分进行传递；

c) *originator* 发起者：启动操作的用户。

4 缩写词

本建议使用下列缩写词：

AVA 属性值判断语
DIB 号码簿信息库
DIT 号码簿信息树
DMD 号码簿管理域
DSA 号码簿系统代理
DUA 号码簿用户代理
RDN 相对可识别名

5 规约

本建议使用建议 X.407 定义的抽象服务定义规约。

第 2 部分 — 抽象服务

6 号码簿服务概述

6.1 如建议 X.501 所述，号码簿服务是通过访问点向代表用户的 DUA 提供的。这些概念在图 1/X.511 中得以描述。

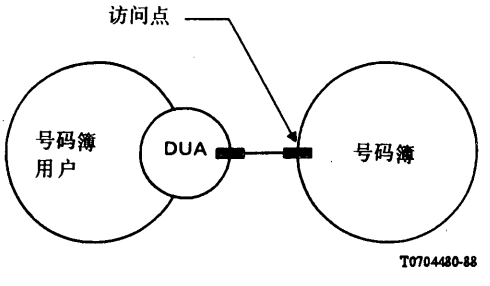


图 1/X.511
访问号码簿

6.2 原则上，至号码簿的访问点可以有几种类型，每一种类型提供不同的服务组合。不妨把号码簿看成为支持各种类型端口的一个客体。每一个端口定义一种特定的交互式操作，通过这个端口号码簿能够同 DUA 进行这种交互式操作。每一个访问点对应一个特定的端口类型组合。

6.3 使用建议 X.407 定义的符号，号码簿可定义如下：

```
directory
    OBJECT
        PORTS { readPort [S],
                searchPort [S],
                modifyPort [S]}
::=  id-ot-directory
```

号码簿通过下列端口提供各种操作：读端口支持用户从 DIB 中的某一个被指名登录项中读取信息；搜索端口允许对 DIB 进行更深入的查阅；修改端口使用户能够修改 DIB 中的登录项。

注 — 预期将来还会有其他类型号码簿端口。

6.4 同样的，DUA（从号码簿的观点来看）可定义如下：

```
dua
    OBJECT
        PORTS { readPort [C],
                searchPort [C],
                modifyPort [C]}
::=  id-ot-dua
```

DUA 使用号码簿提供的服务。

6.5 §6.2 至 §6.4 所述的端口可定义如下：

```
readPort
    PORT
        CONSUMER INVOKES {
            Read, Compare, Abandon}
::=  id-pt-search

searchPort
    PORT
        CONSUMER INVOKES {
            List, Search}
::=  id-pt-search

modifyPort
    PORT
    CONSUMER INVOKES {
        AddEntry, RemoveEntry,
        ModifyEntry, ModifyRDN}
    ::=  id-pt-modify
```

6.6 读端口 **readPort**，搜索端口 **searchPort** 和修改端口 **modifyPort** 操作分别在 §9、§10 和 §11 中定义。

6.7 这些端口只作为用来描述号码簿服务的一种方法。号码簿操作的一致性要求在建议 X.519 中给出。

7 信息类型

7.1 简介

7.1.1 本段识别和定义在号码簿操作的定义中将要使用的一些信息类型。这些信息类型是多种操作所共享

的，是将来可能要使用的，或者由于复杂或单独形成一个整体以致有必要根据使用它们的操作分别定义。

7.1.2 号码簿服务定义使用的一些信息类型在其他有关的建议中定义。7.2 节用于识别类型和指出定义的来源。其余每一节（§7.3 至§7.10）识别和定义一种信息类型。

7.2 在其他建议中定义的信息类型

7.2.1 下列信息类型在建议 X.501 中定义：

a) **Attribute** 属性；
b) **AttributeType** 属性类型；
c) **AttributeValue** 属性值；
d) **AttributeValueAssertion** 属性值判断语；
e) **DistinguishedName** 可识别名；
f) **Name** 名称；
g) **RelativeDistinguishedName** 相对可识别名。

7.2.2 下列信息类型在建议 X.520 中定义：

a) **PresentationAddress** 表示地址。

7.2.3 下列信息类型在建议 X.509 中定义：

a) **Certificate** 证件；
b) **SIGNED** 签署；
c) **CertificationPath** 凭证途径。

7.2.4 下列信息类型在建议 X.214 中定义：

a) **InvokeID** 调用标识符。

7.2.5 下列信息类型在建议 X.518 中定义：

a) **OperationProgress** 操作进展；
b) **ContinuationReference** 继续参考。

7.3 一般自变量

7.3.1 **CommonArguments** 信息可用来说明号码簿所能完成的每一个操作的内容。

```
CommonArguments ::=    SET {
    [30] ServiceControls DEFAULT { },
    [29] SecurityParameters DEFAULT { },
    requestor [28] DistinguishedName
                              OPTIONAL,
    [27] OperationProgress DEFAULT notStarted,
    aliasedRDNs [26] INTEGER OPTIONAL,
    extensions [25] SET OF EXTENSION OPTIONAL}

Extension ::=    SET {
    identifier      [0] INTEGER,
    critical        [1] BOOLEAN DEFAULT FALSE,
    item            [2] ANY DEFINED BY identifier}
```

7.3.2 各成分的意义在§7.3.2.1 至§7.3.2.4 中定义。

7.3.2.1 **ServiceControls** 在§7.5 中说明。它的缺席表示控制集合为空。

7.3.2.2 **SecurityParameters** 在§7.9 中说明。它的缺席表示保密参数集合为空。

7.3.2.3 **requestor DistinguishedName** 识别特定抽象操作的发起者。它包含在与号码簿耦合的一段时间里

获知的用户名。当需要签署该请求时（见§7.10）可对它提出要求，并且应包含发起者的用户名。

7.3.2.4 **OperationProgress** 定义DSA在对请求进行分布式鉴别时所起的作用。在建议X.518中对其有更全面的定义。

7.3.2.5 **aliasedRDNs** 向DSA指出，操作的客体成分是在先前进行操作试探时通过更换别名而产生的。整数值指出在更换别名时产生的客体的RDN数。（该值是在先前的操作中发送转移参考响应时指定的。）

7.3.2.6 **extensions** 为号码簿抽象操作的自变量形式提供表示标准扩展的机制。

注 — 这样的扩展的抽象操作的结果形式是与非扩展的结果形式相同的。（但是，某些扩展的抽象操作的结果有可能不同于其非扩展的抽象操作的结果）。

子成分在§7.3.2.6.1至§7.3.2.6.3中定义。

7.3.2.6.1 **identifier** 用来识别某一个扩展。这个成分的值仅在本建议系列的新版本里分配。

7.3.2.6.2 **critical** 子成分允许扩展的抽象操作的发起者指出只有抽象操作扩展形式的性能可以接受（即，非扩展形式是不能接受的）。在此情况下，扩展是*关键性扩展*。倘若号码簿，或它的某一部分，不能完成关键性扩展，则它应回送一个指示**unavailableCritical Extension**（做为一个**ServiceError**或**Partial-OutcomeQualifier**）。倘若号码簿不能完成非关键性扩展，则它可忽略扩展的存在。

7.3.2.6.3 子成分**item**为号码簿提供完成抽象操作扩展形式所需要的信息。

7.4 *一般结果*

7.4.1 **CommonResults**信息应当存在，以便说明号码簿每次能够完成的检索操作的结果。

```
CommonResults ::= SET {
    [30] SecurityParameters      OPTIONAL,
    performer [29] DistinguishedName
                                 OPTIONAL,
    aliasDereferenced [28]       BOOLEAN
                        DEFAULT FALSE}
```

7.4.2 各成分的含义在§7.4.2.1至§7.4.2.3中定义。

7.4.2.1 **SecurityParameters**在§7.9中说明。它的缺席表示保密参数集合为空。

7.4.2.2 **performer DistinguishedName**识别某一个操作的完成者。当需要签署其结果时，可能要用到它（见§7.10）；它应包含签署该结果的DSA的名称。

7.4.2.3 **aliasDereferenced**：当操作目标客体或基本客体的声称名包含别名（被更换的）时，就把它置成真（**TRUE**）。

7.5 *服务控制*

7.5.1 **ServiceControl**参数（若有的话）包含用来支配或约束服务的控制。

```
ServiceControls      ::=      SET {
    options [0] BIT STRING {
        preferChaining(0)
        chainingProhibited (1),
        localScope (2),
        dontUseCopy (3),
        dontDereferenceAliases(4)}
        DEFAULT {},

priority [1] INTEGER {
    low (0),
    medium (1),
    high (2) } DEFAULT medium,
```

```
timeLimit [2]    INTEGER OPTIONAL,
sizeLimit  [3]   INTEGER OPTIONAL,
scopeOfReferral [4] INTEGER {
                        dmd(0),
                        country(1)}
                        OPTIONAL }
```

7.5.2 各成分的含义在§7.5.2.1至§7.5.2.5中定义。

7.5.2.1 **options** 包含许多指示，每一个指示（当被置位时）肯定所建议的条件：

a) **preferChaining** 表示首先选择使用连锁方式而不是转移参考来提供服务。号码簿不一定要遵照这种选择；

b) **ChainingProhibited** 表示不允许使用连锁方式以及其他把请求分布于号码簿周围的方式；

c) **localScope** 表示本操作只限于在本地范围内完成。这个任选项本身属于本地决策。例如，在一个DSA或一个DMD范围内；

d) **dontUseCopy** 表示不应当使用拷贝信息（在建议X.518中定义）来提供服务；

e) **dontDereferenceAliases** 表示凡是用来识别受操作影响的登录项的别名都不应换名；

注 — 目的是为了能够查阅某一个别名登录项本身而不是包含别名的登录项，如为了读别名登录项。

倘若上述成分被删除，则认为：不选择连锁方式，但也不禁止使用连锁方式；操作范围不加限制；允许使用拷贝；别名将换名（除修改操作之外，修改操作都不更换别名）。

7.5.2.2 **priority**（低、中或高）表示提供服务的优先级。注意，这不是一个有保证的服务，因为号码簿基本上不实现排队。这与在底层使用“优先级”毫无关系。

7.5.2.3 **timeLimit** 表示延迟提供服务的最大时限，单位为秒。若不能满足时限要求，则应报告差错。若省略此成分，则表示不加时限。在 ***List*** 或 ***Search*** 上超过时限的情况下，其结果是（任意）选择累积结果。

注 — 这个成分不包含在时间消逝的过程中处理请求所花的时间：在时间消逝的过程中，有可能有任意个DSA参与对请求的处理。

7.5.2.4 **sizeLimit** 只适用于 **List** 和 **Search** 操作。它表示应回送的客体的最大数目。当超过限额时，**List** 和 **Search** 的操作结果有可能变成任意选择累积结果，在数量上相当于限额。多余的结果应放弃。

7.5.2.5 **scopeOfReferral** 表示DSA回送转移参考的适当范围。视是否选择 **dmd** 值或 **country** 而定，只有指向选择范围内的其他DSA的转移参考才有可能被回送。

上述规定适用于 **List** 和 **Search** 操作结果中的 **ReferralError** 和 **unexplored** 参数。

7.5.3 某些由 **priority**，**timeLimit** 和 **sizeLimit** 组成的组合有可能产生矛盾。例如，一个短时限可能和低优先级发生冲突；一个高限额可能和短时限发生冲突，等等。

7.6 登录项信息选择

7.6.1 **EntryInformationSelection** 参数表示在检索服务中要求提供的登录项信息。

```
EntryInformationSelection ::= SET {
    attributeTypes
        CHOICE {
                allAttributes [0] NULL,
                select [1]] SET OF AttributeType
                -- empty set implies no attributes
                -- are requested --}
        DEFAULT allAttributes NULL,

    InfoTypes [2] INTEGER {
        attributeTypesOnly (0),
        attributeTypesAndValues (1) }
        DEFAULT attributeTypesAndValues }
```

7.6.2 各成分的含义在§7.6.2.1和§7.6.2.2中定义。

7.6.2.1 **attributeTypes** 表示所要求的信息的属性集合：

a) 若选择 **select** 任选，则列出有关的属性。若给出空表，则不回送任何属性。若所选择的属性存在，则应回送该属性信息。仅当所选择的属性一个也不存在时，才回送带有 **noSuchAttribute** 问题的 **AttributeError**。

b) 若选择 **allAttributes** 任选，则必须提供有关登录项的所有属性的信息。

仅当访问权充分时才回送属性信息。仅当访问权不包括查阅全部属性值的权力时才回送 **SecurityError**（有 **insufficientAccessRights** 问题）。

7.6.2.2 **infoTypes** 表示是要求同时提供属性类型和属性值信息（默契值），或是只要求提供属性类型信息。倘若 **attributeTypes**（§7.6.2.1）不要求提供任何属性，则此成分无效。

7.7 登录项信息

7.7.1 **EntryInformation** 参数的功能是从一个登录项中提供所选择的信息。

```
EntryInformation    ::=    SEQUENCE {
    DistinguishedName,
    fromEntry BOOLEAN DEFAULT TRUE,
    SET OF CHOICE {
        AttributeType,
        Attribute} OPTIONAL }
```

7.7.2 登录项的 **DistinguishedName** 始终不能空缺。

7.7.3 **fromEntry** 参数表示信息是从登录项（**TRUE**）抑或登录项的一个拷贝（**FALSE**）中获得的。

7.7.4 在适当情况下可以包括 **AttributeTypes** 或 **Attributes**，它们可单独存在或附带一个或多个属性值。

7.8 过滤器

7.8.1 **Filter** 参数对登录项进行测试；测试结果是某一个登录项满足或不满足该测试条件。过滤器是用判断语表示的，判断登录项的某些属性值或该属性是否存在，而且仅当判断结果为 **TRUE** 时该过滤器才得到满足。

注 — 过滤器可能为 **TRUE**，**FALSE** 或无定义。

```
Filter    ::=    CHOICE {
    item         [0]    FilterItem,
    and          [1]    SET OF Filter,
    or           [2]    SET OF Filter,
    not          [3]    Filter }
```

```
FilterItem ::=          CHOICE {
    equality            [0]    AttributeValueAssertion,
    substrings          [1]    SEQUENCE {
        type                   AttributeType,
        strings SEQUENCE OF CHOICE {
                Initial    [0]    AttributeValue,
                any        [1]    AttributeValue,
                final      [2]    AttributeValue}},
    greaterOrEqual      [2]    AttributeValueAssertion,
    lessOrEqual         [3]    AttributeValueAssertion,
    present             [4]    AttributeType,
    approximateMatch    [5]    AttributeValueAssertion }
```

7.8.2 **Filter** 是一个 **FilterItem**（见 § 7.8.3），或是包含一些简单 **Filter** 的表达式；该简单 **Filter** 是使用逻辑算子 **and**，**or** 和 **not** 组合的。若 **Filter** 是一个未定义的 **FilterItem**，或包含一个或多个未定义的简单 **Filter**，则该 **Filter** 也是无定义。在其他情况下，若 **Filter** 为：

a） 一个 **item**，则仅当对应的 **FilterItem** 为 **TRUE** 时该 **Filter** 才为 **TRUE**；

b） 一个 **and**，则除非嵌套 **Filter** 中有任何一个取 **FALSE**，该 **Filter** 应为 **TRUE**；

注 — 因此，若无嵌套 **Filter**，则 **and** 应取 **TRUE** 值。

c） 一个 **or**，则除非嵌套 **Filter** 中有任何一个取 **TRUE**，该 **Filter** 应为 **FALSE**；

注 — 因此，若无嵌套 **Filter**，则 **or** 应取 **FALSE** 值。

d） 一个 **not**，则仅当嵌套 **Filter** 为 **FALSE** 时该 **Filter** 才为 **TRUE**。

7.8.3 **FilterItem** 是一个判断语，用来判断被测登录项的某一个类型的属性是否存在或取何值。判断语有 **TRUE**，**FALSE** 和无定义三种。

7.8.3.1 每一个 **FilterItem** 包括一个 **AttributeType**，用来表示有关的属性。

7.8.3.2 上述属性值的判断语仅当 **AttributeType** 为已知，而且声称的 **AttributeValue** 符合属性类型的属性语法时才有定义。

注 1 — 在上述条件得不到满足的情况下，**FilterItem** 是无定义的。

注 2 — 访问控制的限制条件有可能要求把 **FilterItem** 视为无定义。

7.8.3.3 属性值的判断语，是根据该属性类型的属性语法定义的匹配规则取值的。非根据特定属性语法定义的匹配规则，不能用于判断该属性。

注 — 在上述条件得不到满足的情况下，**FilterItem** 是无定义的。

7.8.3.4 **FilterItem** 有可能无定义（如 § 7.8.3.2 和 § 7.8.3.3 所述）。在其他情况下，当 **FilterItem** 取下列判断语时情况分别如下：

a） **equality**：仅当有一个属性值等于被判断的属性值时，**FilterItem** 才取 **TRUE**；

b） **substrings**：仅当有一个属性值，其中有关的子串按给定顺序出现时，**FilterItem** 才取 **TRUE** 值。子串不应当有任何重叠，而且子串与最后的属性值之间和子串和子串之间可以（但不一定要）插入

零个或多个串元素。

倘若 **initial** 存在，则子串应与属性值的初始子串相匹配；倘若 **final** 存在，则子串应与属性值的最后一个子串相匹配；倘若 **any** 存在，则子串应与属性值中的任何子串相匹配；

c) **greaterOrEqual**： 仅当相对排序（如适当排序算法所定义）把给出的值排在属性的任意值之前，或等于该属性值时，**FilterItem** 才取 **TRUE** 值；

d) **lessOrEqual**： 仅当相对排序（如适当排序算法所定义）把给出的值排在该属性的任意值之后，或等于该属性值时，**FilterItem** 才取 **TRUE** 值；

e) **present**： 仅当该属性存在于登录项中时，**FilterItem** 才取 **TRUE** 值；

f) **approximateMatch**： 仅当有一个属性值与本地定义的近似匹配算法（例如，拼音变化、语音匹配等）判断结果相配时，**FilterItem** 才取 **TRUE** 值。本建议的当前版本不提供有关近似匹配的具体准则。倘若不支持近似匹配，则应把 **FilterItem** 当做等同（**equality**）匹配看待。

7.9 保密参数

7.9.1 **SecurityParameter** 控制着与号码簿操作有关的各种保密措施。

注 — 这些参数由发送者传递给接收者。当该参数出现在某一个抽象操作的自变量时，请求者为发送者，而执行者则为接收者。在结果中，上述角色的位置正好相反。

```
SecurityParameters  ::=          SET {
    certification-path           [0]
    CertificationPath        OPTIONAL,
    name          [1]        DistinguishedName
                                 OPTIONAL,
    time          [2]        UTCTime OPTIONAL,
    random        [3]        BIT STRING OPTIONAL,
    target        [4]        ProtectionRequest OPTIONAL
                                          }

ProtectionRequest   ::=      INTEGER {
                                 none(0),
                                 signed (1)}
```

7.9.2 各个成分的含义在§7.9.2.1至§7.9.2.5中定义。

7.9.2.1 **CertificationPath** 包含发送者的证件和必要时一序列证件对。证件是用来联系发送者的公用钥匙和可识别名，并且可用来验证自变量或结果中的签名。倘若已在自变量或结果上签名，则此参数应当存在。证件对序列是由发证机构的交叉证件组成的。这个交叉证件是用来验证发送者的证件。当接收者和发送者共享同一个发证机构时就不需要使用该交叉证件。倘若接收者要求提供有效的证件对，而此参数不存在，则该接收者是否拒绝在该自变量或结果上签名，还是设法建立一条凭证途径，均属本地决策。

7.9.2.2 **name** 是自变量或结果的第一个预期的接收者的可识别名。例如，当 **DUA** 生成一个署名的自变量时，该名便是参与操作的 **DSA** 的可识别名。

7.9.2.3 **time**：在使用署名的自变量时，**time** 表示签名的有效时间。这个时间常与随机数结合使用，以便能够检测重播的开始。

7.9.2.4 **random** 是一个数，它在每一个未过期的标志中应当不同。此参数常与时间参数结合使用，以便在签署自变量或结果之后能够检测重播的开始。

7.9.2.5 **target ProtectionRequest** 只可出现于将执行的操作请求中，并指出请求者对结果所希望获得的保密度。可以提供两种保密性能：**none**（不要求保密）和 **signed**（要求号码簿签署该结果，默认值）。对结果实际提供的保密度是由结果的形式指出的，根据号码簿条件的限制，提供的保密度可以相等于或低于请求的等级。

7.10 任选签署

7.10.1 **OPTIONALLY-SIGNED** 是根据生成者的任意选择，其值可包含数字式签名的一个信息类型。这个性能用宏描述如下：

```
OPTIONALLY-SIGNED MACRO ::=
BEGIN

TYPE NOTATION    ::=     type (Type)
VALUE NOTATION ::=       value (VALUE
        CHOICE   { Type, SIGNED Type})

END
```

7.10.2 说明信息签署 **SIGNED** 形式的宏在建议 X.509 中描述。

8 耦合与释放操作

DirectoryBind 和 **DirectoryUnbind** 操作分别在 § 8.1 和 § 8.2 中描述，它们是供 **DUA** 在访问号码簿的开始和结束阶段使用的。

8.1 号码簿耦合

8.1.1 **DirectoryBind** 操作是在访问号码簿的开始阶段使用的。

```
DirectoryBind          ::=     ABSTRACT-BIND
        TO { readPort, searchPort, modifyPort }
        BIND
        ARGUMENT  DirectoryBindArgument
        RESULT    DirectoryBindResult
        BIND-ERROR DirectoryBindError

DirectoryBindArgument     ::=     SET {
        credentials     [0]     Credentials  OPTIONAL,
        versions        [1]     Versions DEFAULT
                                        v1988}

Credentials     ::=     CHOICE {
        simple          [0]     SimpleCredentials,
        strong          [1]     StrongCredentials,
        externalProcedure [2] EXTERNAL }

SimpleCredentials       ::=     SEQUENCE {
        name            [0]     DistinguishedName,
        validity        [1]     SET {
            time1       [0]     UTCTime OPTIONAL,
            Time2       [1]     UTCTime OPTIONAL,
            random1     [2]     BIT STRING OPTIONAL,
            random2     [3]     BIT STRING OPTIONAL } OPTIONAL,
            -- in most instances the argument for
            -- time and random are relevant in
            -- dialogues employing protected password
            -- mechanisms and derive their meaning
            -- as per bilateral agreements

password  [2]           OCTET STRING OPTIONAL }
        -- the value could be an unprotected
        -- password or Protected1 or Protected2
        -- as specified in Recommendation X.509.

StrongCredentials       ::=     SET {
        certification-path[0]  CertificationPath
                                        OPTIONAL,
        bind-token      [1]     Token }
```

```
Token            ::=   SIGNED SEQUENCE {
      algorithm        [0]    AlgorithmIdentifier,
      name             [1]    DistinguishedName,
      time             [2]    UTCTime,
      random           [3]    BIT STRING }

Versions         ::=   BIT STRING {v1988(0)}

DirectoryBindResult ::=     DirectoryBindArgument

DirectoryBindError          ::=          SET {
      versions  [0]   Versions DEFAULT v1988,
      CHOICE {
          serviceError      [1]      ServiceProblem
          securityError     [2]      SecurityProblem
}}
```

8.1.2 各成分的含义在§8.1.2.1至§8.1.2.2中定义。

8.1.2.1 **DirectoryBindArgument** 的 **Credentials** 允许号码簿建立用户标识符。**Credentials** 可能是 **Simple**，**strong**（如建议 X.509 所述）或外部定义（**externalprocedure**）。

8.1.2.1.1 **SimpleCredentials** 是由一个名称（永远是某一个客体的可识别名）和一个（任选）口令组成的。这个简单凭证提供有限的安全保障。倘若口令是建议 X.509 §5 所述的加密口令，则 **SimpleCredentials** 包含名称、口令、和（任选）时间和/或用来检测重播的随机数。在某些情况下，加密口令是由一个客体核对的；该客体只有在本地重新生成用来保护自己的口令拷贝的密码和使用耦合自变量（口令）的值来计算结果之后才知道该口令。在其他情况下，有可能采用直接对比法。

8.1.2.1.2 **StrongCredentials** 是由一个耦合标志、任选的证件和发证机构交叉证件序列（如建议 X.509 所述）组成的。有了这个自变量，号码簿就能够验证正在建立关联的请求的身份证，反之亦然。耦合标志自变量的用法如下：**algorithm** 是用来签署信息的算法标识符；**name** 是预期的接受者的名字。**time** 参数包含标志的有效时间。**random** 随机数是这样的一个数，它在每一个未过期的标志中的值应当不同，并且接受者可以用它来检测重播的开始。

8.1.2.1.3 倘若使用 **externalProcedure**，则正在使用的验证方案的语义学不属于本号码簿的范围。

8.1.2.2 **DirectoryBindArgument** 中的自变量 **Versions** 表示 DUA 准备参与使用的服务版本。对于本协议的当前版本而言，其值应置为 **v1988**（0）。

8.1.2.3 为了便于将来过渡到号码簿的新版本，必须遵照下列规定：

a) 除本建议定义的 **DirectoryBindArgument** 的要素之外，任何要素都应当被接受和不予理睬；

b) 倘若 **DirectoryBindArgument**（例如 **Versions**）的命名比特的附加任选项未定义，则应被接受或不予理睬。

8.1.3 倘若耦合请求获得成功，则应回送一个结果。结果参数的含义在§8.1.3.1与§8.1.3.2中定义。

8.1.3.1 **DirectoryBindResult** 中的 **Credentials** 允许用户建立 DSA 的识别符。这些凭证允许将用来识别 DSA（即直接提供号码簿服务的 DSA）的信息传递给 DUA。这些凭证的形式（即 **CHOICE**）必须与用户所提供的相同。

8.1.3.2 **Directory Bind Result** 中的参数 **Versions** 表示 DSA 实际要向 DUA 提供的服务版本。

8.1.4 倘若耦合请求失败，则应回送在§8.1.4.1和§8.1.4.2中定义的耦合差错。

8.1.4.1 **DirectoryBindError** 中的参数 **Versions** 表示该 DSA 所支持的版本。

8.1.4.2 **SecurityError** 或 **ServiceError** 应提供如下：

- **securityError** **inappropriateAuthentication**
  invalidCredentials
- **serviceError** **unavailable.**

8.2 号码簿的释放

8.2.1 **DirectoryUnbind** 操作是在访问号码簿的结束阶段使用的。

```
DirectoryUnbind    ::=    ABSTRACT-UNBIND
    FROM {readPort, searchPort, modifyPort }
```

8.2.2 **DirectoryUnbind** 不含自变量。

9 号码簿读操作

号码簿有两种类似读操作：**Read** 和 **Compare**，分别在§9.1和§9.2中定义。为了方便起见，在§9.3中定义的 **Abandon** 操作同 **Read** 操作合并在一起。

9.1 读

9.1.1 读操作是用来从一个明显标识的登录项中提取信息。此外，它还可以用来验证一个可识别名。操作自变量是否要签署，可由请求者任意选择（见§7.10）。倘若要求签署，则号码簿可以在结果上签字。

```
Read ::= ABSTRACT-OPERATION
    ARGUMENT          ReadArgument
    RESULT ReadResult
    ERRORS {
        AttributeError, NameError,
        ServiceError, Referral, Abandoned,
        SecurityError }

ReadArgument ::=    OPTIONALLY-SIGNED SET {
    object          [0]  Name,
    selection       [1]  Selection F13 EntryInformationSelection
                         DEFAULT {}
    COMPONENTS OF CommonArguments }

ReadResult    ::=   OPTIONALLY-SIGNED SET {
    entry           [0]  EntryInformation,
    COMPONENTS OF CommonResults }
```

9.1.2 各自变量的含义在§9.1.2.1至§9.1.2.3中定义。

9.1.2.1 自变量 **object** 识别客体登录项；该登录项包含所请求的信息。当名称包含一个或多个别名时，应更换这些别名（除非有关的服务控制禁止这样做）。

9.1.2.2 自变量 **selection** 表示要从登录项中请求什么信息（见§7.6）。

9.1.2.3 **CommonArguments**（见§7.3）包含适合于请求的服务控制的说明。在本操作中，**sizeLimit** 不适用，因此可以忽略。

9.1.3 倘若请求获得成功，则应回送结果。结果参数的含义在§9.1.3.1和§7.4中定义。

9.1.3.1 **entry** 结果参数包含所请求的信息（见§7.7）。

9.1.4 倘若请求失败，则应报告所列差错之一。当在 **selection** 中没有明显列出可以回送的属性时，可以报告带 **noSuchAttribute** 问题的 **AttributeError**。在其他情况下给出的差错报告在§12中定义。

9.2 比较

9.2.1 **Compare** 操作是用来将某一个值（即请求的一个自变量）和某一客体登录项中的某一属性类型的一

个或多个值进行比较。操作自变量可由请求者任意选择是否要加以签署（见§7.10）。若请求签署，则号码簿可签署该结果。

```
Compare ::=        ABSTRACT-OPERATION
      ARGUMENT       CompareArgument
      RESULT     CompareResult
      ERRORS {
          AttributeError, NameError,
          ServiceError, Referral, Abandoned,
          SecurityError }

CompareArgument ::=     OPTIONALLY-SIGNED
SET {
      object        [0]     Name,
      purported     [1]     AttributeValueAssertion,
      COMPONENTS OF CommonArguments }

CompareResult     ::=     OPTIONALLY-SIGNED
SET {
      DistinguishedName OPTIONAL,
      matched       [0]     BOOLEAN,
      from Entry    [1]     BOOLEAN DEFAULT TRUE,
      COMPONENTS OF CommonResults }
```

9.2.2 各自变量的含义在§9.2.2.1至§9.2.2.3中定义。

9.2.2.1 自变量 **object** 是某一有关客体登录项的名称。当 **Name** 包含一个或多个别名时，这些别名应加以更换（除非有关的服务控制禁止这样做）。

9.2.2.2 自变量 **purported** 表示属性类型以及要和登录项中的值相比较的值。

9.2.2.3 **CommonArguments**（见§7.3）表示适合于请求的服务控制。本操作与 **sizeLimit** 无关，故可忽略。

9.2.3 倘若请求获得成功(即实际完成比较功能)，则应回送结果。结果参数的含义在§9.2.3.1、§9.2.3.2和§7.4中定义。

9.2.3.1 当有别名已被更换并代表客体本身的可识别名时，则 **DistinguishedName** 存在。

9.2.3.2 结果参数 **matched** 表示比较的结果。当对各值进行了比较，且相匹配时，该参数取 **TRUE** 值，否则应取 **FALSE**。

9.2.3.3 倘若 **fromEntry** 为 **TRUE**，则应把信息与登录项加以比较；倘若为 **FALSE**，则应把信息的一部分与拷贝加以比较。

9.2.4 倘若请求失败，应报告列出的差错之一。§12中定义了报告差错的各种情况。

9.3 *放弃*

9.3.1 在号码簿查询操作中，当用户对结果不再感兴趣时，可以借助于 **Abandon** 放弃该操作。

```
Abandon     ::=    ABSTRACT-OPERATION
      ARGUMENT       AbandonArgument
      RESULT     AbandonResult
      ERRORS {AbandonFailed}

AbandonArgument  ::=   SEQUENCE {
      InvokeID      [0]  InvokeID}

AbandonResult       ::=  NULL
```

9.3.2 有一个单独的自变量 **InvokeID**，用以表示将要放弃的操作。该 **invokeID** 的值与曾用来调用将放弃的操作的 **invokeID** 相同。

9.3.3 当请求获得成功时，应回送一个结果，虽然不必传递任何信息。原操作将由于 **Abandoned** 差错而宣告失败。

9.3.4 当请求失败时，应报告 **Abandon Failed** 差错。这种差错在§12.3中说明。

9.3.5 **Abandon** 仅适用于查询操作，如 **Read**，**compare**，**List** 和 **Search**。

9.3.6 一个 DSA 可在本地放弃操作。倘若 DSA 已把该操作连到，或扩散到其它各 DSA，则该 DSA 可以要求它们放弃该操作。DSA 可选择不放弃操作，从而应回送 **AbandonFailed** 差错。

10 号码簿搜索操作

有两种“类似搜索”操作：**List** 和 **Search**，分别在§10.1和§10.2中定义。

10.1 列表

10.1.1 **List** 操作是用来获得显登录项的直接下级名单。在某些情况下，回送的名单可能不完整。操作自变量可由请求者选择是否要签署（见§7.10）。倘若要求签署，则号码簿可签署其结果。

```
List ::=        ABSTRACT-OPERATION
      ARGUMENT          ListArgument
      RESULT       ListResult
      ERRORS {
                        NameError
            ServiceError, Referral, Abandoned,
            SecurityError }

List Argument  ::=          OPTIONALLY-SIGNED SET {
      object    [0]         Name,
      COMPONENTS OF CommonArguments }

ListResult       ::=        OPTIONALLY-SIGNED
CHOICE {
      listInfo SET {
      DistinguishedName OPTIONAL,
      subordinates [1]       SET OF SEQUENCE {
          RelativeDistinguishedName,
          aliasEntry [0]     BOOLEAN DEFAULT FALSE
          fromEntry  [1]     BOOLEAN DEFAULT TRUE},

      partialOutcomeQualifier [2]
                                 PartialOutcomeQualifier     OPTIONAL
      COMPONENTS OF CommonResults },
      uncorrelatedListInfo     [0] SET OF
                               ListResult }

PartialOutcomeQualifier ::=    SET {
      limitProblem     [0]    LimitProblem
            OPTIONAL,
      unexplored       [1]    SET OF
            ContinuationReference   OPTIONAL,
      unavailableCriticalExtensions [2] BOOLEAN DEFAULT FALSE }

LimitProblem ::=  INTEGER {
      timeLimitExceeded (0),
      sizeLimitExceeded (1),
      administrativeLimitExceeded (2) }
```

10.1.2 各自变量的含义在§10.1.2.1和§7.3中定义。

10.1.2.1 自变量 **object** 表示客体登录项（或可能是根），该客体登录项的直接下级将列入表中。当 **Name** 包含一个或多个别名时，应换名（除非有关的服务控制禁止这样做）。

10.1.3 只要能找到客体，则不管是否有可以回送的下级信息，请求都是成功的。结果参数的含义在§10.1.3.1至§10.1.3.4和§7.4中定义。

10.1.3.1 **DistinguishedName** 应当在更换别名的情况下存在。它代表客体本身的可识别名。

10.1.3.2 参数 **subordinates** 传送有关被指名登录项的直接下级的信息（如果有的话）。倘若下级登录项为别名，则它们不应当换名。

10.1.3.2.1 **RelativeDistinguishedName** 属于下级。

10.1.3.2.2 参数 **fromEntry** 指出信息是从登录项（**TRUE**）还是从登录项的一个拷贝（**FALSE**）中获得的。

10.1.3.2.3 参数 **aliasEntry** 指出下级登录项是一个别名登录项（**TRUE**），抑或不是（**FALSE**）。

10.1.3.3 **PartialOutcomeQualifier** 是由三个子成分组成的，如§10.1.3.3.1至§10.1.3.3.3所定义。在结果不完整的情况下，必须提供此参数。

10.1.3.3.1 参数 **LimitProblem** 指出是否超过了规定的时限，限额或行政限制。回送的内容是那些在达到极限时可以获得的结果。

10.1.3.3.2 参数 **unexplored** 应当在DIT的领域没有被检查的情况下存在。它的信息允许DUA必要时通过访问其他的访问点继续处理List操作。该参数包含 **continuationReferences** 集合（有可能是空的），各包含一个基本客体名（从此客体可以继续进行操作），一个 **OperationProgress** 的适当值和为请求的继续进行而提供的访问点集合。回送的 **ContinuationReferences** 应当在操作服务控制所要求的参考范围内。

10.1.3.3.3 参数 **unavailableCriticalExtensions** 若存在的话，指出在号码簿的某些部分曾缺一个或多个关键性扩展。

10.1.3.4 当DUA提出了对 **signed** 的保护请求时，参数 **uncorrelatedListInfo** 可包含许多组由号码簿的各个组成部分所提供的和签署的结果参数。倘若在链中没有一个DSA能够关联全部结果，则DUA应从各个片段中拼凑实际结果。

10.1.4 倘若请求失败，应报告表中所列差错之一。§12中定义了报告差错的各种情况。

10.2 *搜索*

10.2.1 **Search** 操作是用来搜索特定登录项中的部分DIT，并回送从该登录项中选择的信息。操作自变量可由请求者选择是否要签署（见§7.10）。倘若要求签署，则号码簿可以签署其结果。

```
Search ::=  ABSTRACT-OPERATION
      ARGUMENT     SearchArgument
      RESULT     SearchResult
      ERRORS {
            AttributeError, NameError,
            ServiceError, Referral, Abandoned,
            SecurityError }

SearchArgument ::= OPTIONALLY-SIGNED
SET {
      baseObject          [0]  Name,
      subset              [1]  INTEGER {
            baseObject (0),
            oneLevel(1),
            wholeSubtree(2)}  DEFAULT baseObject,
      filter              [2]  Filter DEFAULT and {}.
```

```
        searchAliases       [3]  BOOLEAN DEFAULT TRUE,
        selection           [4]  EntryInformationSelection DEFAULT {}
            COMPONENTS OF CommonArguments }

SearchResult    ::=         OPTIONALLY-SIGNED
        CHOICE {
        searchInfo SET {
        DistinguishedName OPTIONAL,
        entries  [0]        SET OF EntryInformation,
        partialOutcomeQualifier
            [2]PartialOutcomeQualifier OPTIONAL,
COMPONENTS OF CommonResults },
uncorrelatedSearchInfo [0] SET OF
        SearchResult }
```

10.2.2 各自变量的含义在§10.2.2.1至§10.2.2.3、§10.2.2.5和§7.3中定义。

10.2.2.1 自变量 **baseObject** 表示搜索对象客体登录项（或可能是根）。

10.2.2.2 自变量 **Subset** 指出是否要对下列各项进行搜索：

a) 只有 **base Object**；

b) 只有基本客体的直接下级（**oneLevel**）；

c) 基本客体及其所有的下级（**wholeSubtree**）。

10.2.2.3 自变量 **filter** 是用来从搜索空间中删除不需要的登录项。只有满足过滤器的一些登录项，才回送信息（见§7.8）。

10.2.2.4 取决于 **dontDereferenceAliasesServiceControl** 的设置，别名将在寻找基本客体时进行换名。取决于 **SearchAliases** 参数的设置，在搜索过程中基本客体下级中的别名将换名。当 **searchAliases** 为 **TRUE** 时，应更换别名；若参数为 **FALSE**，则不换名。若 **searchAliases** 参数为 **TRUE**，则搜索将在别名客体的子树中继续进行。

10.2.2.5 自变量 **selection** 表示从登录项中要求提供的信息（见§7.6）。

10.2.3 倘若找到了基本客体，则不管有没有要回送的下级，请求都是成功的。

注 — 其必然结果是：对一个登录项而言，（未过滤的）Search 的结果与设法查询该登录项同一个属性集合的 Read 给出的结果不一定相同。这是因为，当登录项中所选择的属性一个也没有时，后者将回送 **AttributeError**。

结果参数的含义在§10.2.3.1至§10.2.3.4以及§7.3中定义。

10.2.3.1 当别名已换名并代表基本客体的可识别名时，**DistinguishedName** 存在。

10.2.3.2 参数 **entries** 是用来传递（0个或多个）满足过滤器（见§7.5）的登录项的信息。

10.2.3.3 **PartialOutcomeQualifier** 包含两个子成分，如在§10.1.3.4中为 **List** 操作描述的那样。

10.2.3.4 参数 **uncorrelatedSearchInfo** 与§10.1.3.4中对 **uncorrelatedListInfo** 的描述类同。

10.2.4 倘若请求失败，则应报告表中所列差错之一。§12中定义了报告差错的各种情况。

11 号码簿修改操作

号码簿的修改操作有以下四种：**AddEntry**，**RemoveEntry**，**ModifyEntry** 和 **ModifyRDN**，并分别在 § 11.1 至 11.4 中定义。

注 1 — 上述抽象操作使用可识别名来识别目标登录项。

注 2 — **AddEntry**，**RemoveEntry** 和 **ModifyRDN** 操作是否成功取决于 DIB 在号码簿里的物理分布。当操作宣告失败时，应报告 **Update Error** 和 **affectsMultipleDSA** 问题。见建议 X.518。

11.1 增加登录项

11.1.1 **AddEntry** 操作是用来增加 DIT 的叶登录项（客体登录项或别名登录项）。操作自变量可由请求者任意选择是否要加以签署（见 § 7.10）。

```
AddEntry    ::=     ABSTRACT-OPERATION
      ARGUMENT        AddEntryArgument
      RESULT          AddEntryResult
      ERRORS {
            AttributeError, NameError,
            ServiceError, Referral, SecurityError,
            UpdateError }

AddEntryArgument  ::=   OPTIONALLY-SIGNED
SET {
      object          [0]   DistinguishedName,
      entry           [1]   SET OF Attribute,
      COMPONENTS OF CommonArguments }

AddEntryResult ::=  NULL
```

11.1.2 各自变量的含义在 § 11.1.2.1 至 § 11.1.2.3 中定义。

11.1.2.1 自变量 **object** 表示要增加的登录项。为了确保操作获得成功，该客体的直接上级必须存在，并且可通过删除最后的 RDN 成分来确定该直接上级（该 RDN 成分是属于要建立的登录项）。

11.1.2.2 自变量 **entry** 包含属性信息；属性信息和 RDN 的信息构成要建立的登录项。号码簿应当保证登录项符合号码簿体制。倘若被建立的登录项是一个别名，则不需要检查 aliasedObjectName 属性是否确实指向一个有效登录项。

11.1.2.3 **CommonArguments**（见 § 7.3）包含适用于请求的服务控制说明。本操作与 **dontDereferenceAlias** 任选项和 **sizeLimit** 成分无关，故若提供则可忽略。本操作不对别名进行换名。

11.1.3 倘若请求获得成功，则尽管不传递信息，也要回送结果。

11.1.4 倘若请求失败，则应报告表中所列差错之一。§ 12 中定义了报告差错的各种情况。

11.2 删除登录项

11.2.1 **RemoveEntry** 操作是用来删除 DIT 中的叶登录项（客体登录项或别名）。操作自变量可由请求者任意选择是否要加以签署（见 § 7.10）。

```
RemoveEntry ::=     ABSTRACT-OPERATION
      ARGUMENT  RemoveEntryArgument
      RESULT    RemoveEntryResult
      ERRORS {
          NameError,
          ServiceError, Referral, SecurityError,
          UpdateError}
```

```
RemoveEntryArgument ::= OPTIONALLY-SIGNED SET {
        object        [0]     DistinguishedName,
        COMPONENTS OF CommonArguments }

RemoveEntryResult     ::= NULL
```

11.2.2 各自变量的含义在§11.2.2.1和§11.2.2.2中定义。

11.2.2.1 自变量 **object** 表示要删除的登录项。名称中的别名将不换名。

11.2.2.2 **CommonArguments**（见§7.3）包含适用于请求的服务控制说明。本操作与 **dontDereferenceAlias** 任选项和 **sizeLimit** 成分无关，故若提供则可忽略。本操作不要求更换别名。

11.2.3 倘若请求获得成功，则尽管不传递信息，也应回送结果。

11.2.4 倘若请求失败，则应报告所列差错之一。§12中定义了报告差错的各种情况。

11.3 修改登录项

11.3.1 **ModifyEntry** 操作是用来完成登录项的下列修改任务：

a) 增加新属性；　b) 删除属性；　c) 增加属性值；

d) 删除属性值；　e) 替换属性值；　f) 修改别名。

操作自变量可由请求者任意选择是否要加以签署（见§7.10）。

```
ModifyEntry ::=        ABSTRACT-OPERATION
        ARGUMENT  ModifyEntryArgument
        RESULT    ModifyEntryResult
        ERRORS {
                AttributeError, NameError,
                ServiceError, Referral, SecurityError,
                UpdateError }

ModifyEntryArgument ::=        OPTIONALLY-SIGNED SET {
        object        [0]    DistinguishedName,
        changes       [1]    SEQUENCE OF EntryModification,
        COMPONENTS OF CommonArguments }

ModifyEntryResult   ::=      NULL

EntryModification   ::=      CHOICE {
        addAttribute  [0]        Attribute,
        removeAttribute          [1]  AttributeType,
        addValues                [2]  Attribute,
        removeValues  [3]        Attribute }
```

11.3.2 各自变量的含义在§11.3.2.1和§11.3.2.2中定义。

11.3.2.1 自变量 **object** 表示要修改的登录项。名称中的任何别名概不换名。

11.3.2.2 自变量 **Changes** 给出按指定顺序要做的一系列修改。倘若有些修改宣告失败，则应生成一个 **AttributeError**，并且让该登录项留在操作前的状态。即，操作具有可分性的特点（**atomic**）。一系列修改的最终结果不应该违反号码簿体制。但是，个别 **EntryModification** 变化违反号码簿体制的行为是可能的，而且有时是必要的。几种可能出现的修改类型如下：

a) **addAttribute**：表示要加到登录项的新属性，它是由自变量明确指出的。任何增加已经存在的属性的尝试均将导致 **AttributeError**。

ARCHIVES U.I.T. GENEVE

b) **removeAttribute**： 这个自变量(按类型)表示要从登录项中删除的属性。任何删除不存在的属性的尝试均将导致 **AttributeError**。

注 — 倘若在 RDN 中有该属性类型，则不允许进行该操作；

c) **addValues**：表示按自变量中的属性类型给出的属性，并指出要加到该属性的一个或多个属性值。任何增加已经存在的值的尝试均将导致差错。任何把一个值加到不存在的类型的尝试均将导致差错。

d) **removeValues**：表示按自变量中的属性类型给出的属性，并指出要从属性中删除的一个或多个属性值。倘若那些值不存在于属性中，则将导致 **AttributeError**。倘若试图修改客体类别属性，则应回送一个更新差错信息。

注 — 倘若其中的一个值存在于 RDN，则不得进行该操作。

单个 **ModifyEntry** 操作中的值可用 **addValues** 和 **removeValues** 的组合代替。

11.3.2.3 **CommonArguments**（见 §7.3）包含使用于请求的服务控制的说明。本操作与 **dontDereferenceAlias** 任选项和 **sizeLimit** 成分无关，故若提供则可忽略。本操作不更换别名。

11.3.3 倘若请求获得成功，则虽然不传递信息，也应回送结果。

11.3.4 倘若请求失败，则应报告所列差错之一。§12 中定义了报告差错的各种情况。

11.4 修改 *RDN*

11.4.1 **ModifyRDN** 操作是用来改变 DIT 中的叶登录项（客体登录项或别名登录项）的相对可识别名。操作自变量可由请求者任意选择是否要加以签署（见 §7.10）。

```
ModifyRDN ::=        ABSTRACT-OPERATION
      ARGUMENT  ModifyRDNArgument
      RESULT    ModifyRDNResult
      ERRORS {
          NameError,
          ServiceError, Referral, SecurityError,
          UpdateError }
ModifyRDNArgument   ::= OPTIONALLY-SIGNED SET {
      object        [0]   DistinguishedName,
      newRDN        [1]   RelativeDistinguishedName,
      deleteOldRDN  [2]   BOOLEAN DEFAULT FALSE,
      COMPONENTS OF CommonArguments }
ModifyRDNResult     ::=   NULL
```

11.4.2 各参数的含义在 §11.4.2.1 至 §11.4.2.5 中定义。

11.4.2.1 自变量 **object** 表示要修改相对可识别名的登录项。名称中的别名不应更换。直接上级登录项中不存在任何非特定下级参考（见建议 X.518）。

11.4.2.2 **newRDN** 表示登录项的新 RDN。

11.4.2.3 倘若新 RDN 中的属性值原先不存在于登录项中（做为旧 RDN 的一部分或非可识别值），则应当将该属性值加到登录项中。倘若无法增加该属性值，则应回送差错信息。

11.4.2.4 倘若 **deleteOldRDN** 标志被置位，则凡是不在新 RDN 出现的所有旧 RDN 中的属性值均应当删除。倘若此标志未置位，则旧值应当留在登录项（不做为 RDN 的一部分）。若在操作过程中 RDN 的单值属性值发生变化，则应将标志置位。倘若本操作删除了属性的最后一个属性值，则应删除该属性。

11.4.2.5 **Common Arguments**(见§7.3)包含使用于请求的服务控制说明。本操作与 **dontDereferenceAlias** 任选项和 **sizeLimit** 成分无关，故若提供则可忽略。本操作不更换别名。

11.4.3 倘若请求获得成功，则虽然不传递信息，也应回送结果。

11.4.4 倘若请求失败，则应报告所列差错之一。§12 中定义了报告差错的各种情况。

11.4.5 根据本建议的定义，本操作只适用于叶登录项。

12 差错

12.1 差错优先级

12.1.1 本号码簿不能在差错情况下继续操作。

注 1 — 在同一个查询的重复实例中，遇到的第一个差错有可能不一样，因为在处理特定查询的过程中是没有具体的逻辑顺序可循的。例如，对各 DSA 可以按不同顺序进行搜索。

注 2 — 这里规定的差错优先级准则仅适用于本号码簿所提供的整个抽象服务。在考虑号码簿的内部结构时可采用不同的准则。

12.1.2 倘若号码簿同时检测出一个以上差错，则应采用下列表格来确定所要报告的差错。名列前矛的差错，其优先级自然高于其他的差错并应优先报告。

a) **NameError**
b) **UpdateError**
c) **AttributeError**
d) **SecurityError**
e) **ServiceError**

12.1.3 下列差错的优先级并不矛盾：

a) **AbandonFailed**： 因为它只存在于特定操作 **Abandon** 中，它不可能遇到其他差错；
b) **Abandoned**： 倘若在检测出某一个差错时，同时收到 **Abandon** 操作，则不报此差错。在此情况下，与报告所遇到的实际差错一起，还应报告表示 **toolate** 问题的 **AbandonFailed** 差错；
c) **Referral**：不是“真正”的差错，它只表示号码簿已检测到 DUA 必须向另一个访问点提出请求。

12.2 放弃

12.2.1 在任何未完成的号码簿查询操作中（即，**Read**，**Search**，**Compare**，**List**），倘若 DUA 使用适当的 **Invoke ID** 调用 **Abandon** 操作，则可以报告这种结果。

Abandoned ::= ABSTRACT-ERROR -- *not literally an "error"*

12.2.2 此种差错没有与之相关的参数。

12.3 放弃操作宣告失败

12.3.1 **AbandonFailed** 差错报告在试图放弃操作的过程中遇到的问题。

```
AbandonFailed  ::=  ABSTRACT-ERROR
     PARAMETER SET {
            problem    [0] AbandonProblem,
            operation  [1] InvokeID}

AbandonProblem ::= INTEGER
                   noSuchOperation (1),
                   tooLate (2),
                   cannotAbandon (3) }
```

12.3.2 各参数的含义在§12.3.2.1和§12.3.2.2中定义。

12.3.2.1 对遇到的问题已加以说明。可指出的问题如下：

a) **noSuchOperation**：当号码簿不知道拟放弃的操作时（这种可能性是有的，例如根本没有人调用过这种操作，或者号码簿忘记了这件事）；

b) **tooLate**：当号码簿已经对该操作作出响应时；

c) **cannotAbandon**：当试图放弃被禁止的操作（如修改），或不能完成放弃操作时。

12.3.2.2 指出拟放弃的特定操作（起动）。

12.4 属性差错

12.4.1 **AttributeError** 报告与属性有关的问题。

```
AttributeError ::= ABSTRACT-ERROR
      PARAMETER SET {
            object   [0]    Name,
            problems [1]    SET OF SEQUENCE {
               problem      [0]  AttributeProblem,
               type         [1]  AttributeType,
               value        [2]  AttributeValue
                                      OPTIONAL }}

AttributeProblem  ::=  INTEGER {
     noSuchAttributeOrValue (1),
     InvalidAttributeSyntax (2),
     undefinedAttributeType (3),
     InappropriateMatching (4),
     constraintViolation (5)
     attributeOrValueAlreadyExists (6) }
```

12.4.2 各参数的含义在§12.4.2.1和§12.4.2.2中定义。

12.4.2.1 参数 **object** 指出差错发生时做为操作对象的登录项。

12.4.2.2 可以指出一个或多个 **problem**。下列每一个 **problem** 伴随有一个属性 **type**，为了避免混淆必要时还可以伴随一个引起问题的 **value**：

a) **noSuchAttributeOrValue**：被指名的登录项缺少操作自变量的一种指定属性或属性值；

b) **invalidAttributeSyntax**：声称的属性值，做为操作自变量，不符合属性类型的属性语法；

c) **undefinedAttributeType**：提供了未定义的属性类型做为操作自变量。此种差错的产生仅与 **Add**，**Remove**，**Modify** 或 **ModifyRDN** 操作有关；

d) **inappropriateMatching**:试图(例如在过滤器中)使用特定属性类型中未定义的匹配规则;

e) **constraintViolation**:在抽象操作的自变量中有一个属性或属性值不符合建议 **X**.501 或属性定义提出的限制条件(如该值超过允许的最大限额);

f) **attributeOrValueAlreadyExists**:试图增加一个在登录项中已经存在的属性,或在属性中已经存在的值。

12.5 名称中的差错

12.5.1 **NameError** 报告与做为操作自变量的名称有关的差错。

```
NameError ::= ABSTRACT-ERROR
    PARAMETER SET {
        problem [0] NameProblem,
        matched [1] Name}

NameProblem ::= INTEGER {
    noSuchObject (1),
    aliasProblem (2),
    invalidAttributeSyntax (3),
    aliasDereferencingProblem (4) }
```

12.5.2 各参数的含义在§12.5.2.1 和§12.5.2.2 中说明。

12.5.2.1 所遇到的特定问题。可指出的问题如下:

a) **noSuch Object**: 提供的名称与任一客体名不符;

b) **aliasProblem**: 有一个别名已换成与客体无关的名称;

c) **invalidAttributeSyntax**: 名称中的属性类型与其在 AVA 中的属性值不一致;

d) **aliasDereferencingProblem**: 有一别名出现在不允许出现的场合。

12.5.2.2 参数 **matched** 包含 DIT 中的最低登录项(客体或别名)名,该名是经过匹配的,而且是提供名、或者若有一别名已换名——合成名的缩短形式。

注 — 倘若在号码簿操作自变量中提供的名称中出现属性类型和/或值的问题,则应以 **NameError**(有 **invalidAttributeSyntax** 问题)而不是 **AttributeError** 或 **UpdateError** 报告差错。

12.6 转移参考

12.6.1 **Referral** 把服务使用者转移到一个或多个质量更好的访问点,以便更好地完成所要求的操作。

```
Referral ::= ABSTRACT-ERROR  -- not literally an "error"
    PARAMETER SET {
     candidate              [0] ContinuationReference }
```

12.6.2 上述差错有个包含 **ContinuationReference** 的单一参数,它可以用来推进操作(见建议 **X**.518)。

12.7 保密差错

12.7.1 **SecurityError** 报告执行保密操作过程中所遇到的问题。

```
SecurityError ::=     ABSTRACT-ERROR
    PARAMETER SET {
     problem [0] SecurityProblem }

SecurityProblem ::= INTEGER {
    InappropriateAuthentication (1),
    InvalidCredentials (2),
    InsufficientAccessRights (3),
    InvalidSignature (4),
    protectionRequired (5),
    noInformation (6) }
```

12.7.2 上述差错有个单一参数，用来报告所遇到的具体 **problem**。可指出的问题如下：

a) **inappropriateAuthentication**：请求者提供的凭证，其保密级别不符合所要求的保密级别，如要求高可靠性凭证时却仅提供了简单凭证；

b) **invalidCredentials**：提供的凭证无效；

c) **insufficientAccessRights**：请求者无权完成所要求的操作；

d) **invalidSignature**：发现请求的签署无效；

e) **protectionRequired**：由于自变量未经签署，号码簿不愿意完成所要求的操作；

f) **noInformation**：由于没有差错信息，所要求的操作产生保密差错。

12.8 服务差错

12.8.1 **ServiceError** 报告与服务有关的问题。

```
ServiceError ::= ABSTRACT-ERROR
      PARAMETER SET {
            problem [0] ServiceProblem },

ServiceProblem ::= INTEGER {
            busy (1),
            unavailable (2),
            unwillingToPerform (3),
            chainingRequired (4),
            unableToProceed (5),
            invalidReference (6),
            timeLimitExceeded (7),
            administrativeLimitExceeded (8),
            loopDetected (9),
            unavailableCriticalExtension (10),
            outOfScope (11),
            ditError (12) }
```

12.8.2 上述差错有个单一参数，用来报告所遇到的具体问题。可指出的问题如下：

a) **busy**：号码簿（或它的某一部分）现在很忙，不能完成所要求的操作，但过一短时间后即可完成；

b) **unavailable**：号码簿（或它的某一部分）现在不能接受访问；

c) **unwillingToPerform**：号码簿（或它的某一部分）不准备接受这个请求，例如由于消耗太多资源或违反行政权威的政策；

d) **chainingRequired**：号码簿无法接受请求，除非通过连锁操作，但是 **chainingProhibited** 服务控制任选项却禁止这样做；

e) **unableToProceed**：回送此差错的 DSA 没有提供适当名称的行政权威，因此也就无法参予解决名称问题；

f) **invalidReference**：DSA 无法接受 DUA（在 **OperationProgress** 中）提出的请求。这可能是由于使用无效参考所致。

g) **timeLimitExceeded**：号码簿已超过用户在服务控制中规定的时限。没有可回送给用户的部分结果；

i) **loopDetected**：由于内部循环，号码簿无法完成所要求的操作；

j) **unavailableCriticalExtension**：由于缺少一个或多个关键性扩展，号码簿无法完成所要求的操作；

k) **outOfScope**：在请求的范围内没有可用的参考；

l) **ditError**：由于 DIT 一致性问题，号码簿无法完成所要求的操作。

12.9 修改上的差错

12.9.1 **UpdateError** 报告因增加、删除或修改 DIB 中的信息而产生的问题。

```
UpdateError ::= ABSTRACT-ERROR
        PARAMETER SET {
            problem  [0]  UpdateProblem }

UpdateProblem     ::=  INTEGER {
        namingViolation (1),
        objectClassViolation (2),
        notAllowedOnNonLeaf (3),
        notAllowedOnRDN (4),
        entryAlreadyExists (5),
        affectsMultipleDSAs (6),
        objectClassModificationProhibited (7) }
```

12.9.2 上述差错有个单一的 **problem** 参数，用来报告遇到的 **problem**。可指出的问题如下：

a) **namingViolation**：增加或修改的尝试均将违反号码簿体制和建议 X.501 中定义的 DIT 结构准则。即，上述尝试将把某登录项当做一个别名登录项的下级，或把登录项放入 DIT 的客体类别的成员不允许进入的 DIT 领域，或为登录项定义一个 RDN 时输入一个禁止使用的属性类型；

b) **objectClassViolation**：经过修改之后将产生一个与其客体类别给出的定义或建议 X.501 关于客体类别的定义不一致的登录项；

c) **notAllowedOnNonLeaf**：只允许对 DIT 的叶登录项进行操作；

d) **notAllowedOnRDN**：操作结果将影响 RDN（例如删除 RDN 的一个属性）；

e) **entryAlreadyExists**：**AddEntry** 操作试图补充一个本来已经存在的登录项；

f) **affectsMultipleDSAs**：要修改就需要对多个 DSA 进行操作，而这样做是不允许的；

g) **objectClassModificationProhibited**：试图修改客体类别属性的操作。

注 — **UpdateError** 不使用于在 **AddEntry**，**RemoveEntry**，**ModifyEntry** 或 **ModifyRDN** 操作中遇到的属性类型、属性值或超限额问题的报告。上述问题是通过一个 **AttributeError** 报告的。

附 件 A

（附于建议 X.511）

ASN.1 中的抽象服务

本附件是本标准的一部分。

本附件包括所有在本建议中使用的 ASN.1 类型、值和宏的定义，并以 ASN.1 模块 **DirectoryAbstractService** 的形式描述。

```
DirectoryAbstractService {joint-ISO-CCITT ds(5) modules(1) directoryAbstractService(2)}
DEFINITIONS ::=
BEGIN

EXPORTS
      directory, readPort, searchPort, modifyPort,
      DirectoryBind, DirectoryBindArgument,
      DirectoryUnbind,
      Read, ReadArgument, ReadResult,
      Abandon, AbandonArgument, AbandonResult,
      Compare, CompareArgument, CompareResult,
      List, ListArgument, ListResult,
      Search, SearchArgument, SearchResult,
      AddEntry, AddEntryArgument, AddEntryResult,
      RemoveEntry, RemoveEntryArgument, RemoveEntryResult,
      ModifyEntry, ModifyEntryArgument, ModifyEntryResult,
      ModifyRDN, ModifyRDNArgument, ModifyRDNResult,
      Abandoned, AbandonFailed, AttributeError, NameError,
      Referral, SecurityError, ServiceError, UpdateError,
      SecurityParameters;

IMPORTS
      informationFramework, authenticationFramework,
            distributedOperations, directoryObjectIdentifiers
         FROM UsefulDefinitions {joint-iso-ccitt ds(5) modules(1)
                                    usefulDefinitions(0)}

OBJECT, PORT, ABSTRACT-BIND, ABSTRACT-UNBIND,
ABSTRACT-OPERATION, ABSTRACT-ERROR
         FROM AbstractServiceNotation {joint-iso-ccitt mhs-motis(6)
                                       asdc(2) modules(0) notation(1) }

Attribute, AttributeType, AttributeValue, AttributeValueAssertion,
DistinguishedName, Name, RelativeDistinguishedName
         FROM InformationFramework InformationFramework

id-ot-directory, id-ot-dua, id-pt-read, id-pt-search, id-pt-modify
         FROM DirectoryObjectIdentifiers directoryObjectIdentifiers

ContinuationReference, OperationProgress
         FROM DistributedOperations distributedOperations

Certificate, CertificationPath, SIGNED,
PROTECTED, AlgorithmIdentifier
         FROM AuthenticationFramework authenticationFramework
InvokeID,
         FROM Remote-Operations-Notation {joint-iso-ccitt
                                          remoteOperations(4) notation(0)};


-- macro for representing optional signing --

OPTIONALLY-SIGNED MACRO ::=
BEGIN
      TYPE NOTATION    ::= type (Type)
      VALUE NOTATION ::= value (VALUE CHOICE { Type, SIGNED Type})
END


-- objects and ports --


directory
      OBJECT
            PORTS { readPort [S].
                    searchPort [S],
                    modifyPort [S]}

. ::= id-ot-directory
```

```
dua
      OBJECT
            PORTS { readPort [C],
                    searchPort [C]
                    modifyPort [C]}
::= id-ot-dua

readPort
      PORT
            CONSUMER INVOKES {
                  Read, Compare, Abandon}
::= id-pt-read

searchPort
      PORT
            CONSUMER INVOKES {
                  List, Search }
::= id-pt-search

modifyPort
      PORT
            CONSUMER INVOKES {
                  AddEntry, RemoveEntry,
                  ModifyEntry, ModifyRDN}
::= id-pt-modify

-- bind and unbind --

DirectoryBind ::=      ABSTRACT-BIND
      TO { readPort, searchPort, modifyPort }
      BIND
      ARGUMENT  DirectoryBindArgument
      RESULT    DirectoryBindResult
      BIND-ERROR DirectoryBindError

DirectoryBindArgument ::= SET {
      credentials     [0] Credentials OPTIONAL,
      versions        [1] Versions DEFAULT v1988}

Credentials ::=       CHOICE {
      simple    [0] SimpleCredentials,
      strong    [1] StrongCredentials,
      externalProcedure [2] EXTERNAL }

SimpleCredentials ::= SEQUENCE {
      name      [0] DistinguishedName,
      validity [1] SET {
         time1        [0] UTCTime OPTIONAL,
         time2        [1] UTCTime OPTIONAL,
         random1      [2] BIT STRING OPTIONAL,
         random2      [3] BIT STRING OPTIONAL }
                              OPTIONAL,
         password     [2] OCTET STRING OPTIONAL }

StrongCredentials ::= SET {
      certification-path     [0] CertificationPath OPTIONAL,
      bind-token             [1] Token }

Token ::= SIGNED SEQUENCE {
      algorithm   [0]   AlgorithmIdentifier
      name        [1]   DistinguishedName,
      time        [2]   UTCTime,
      random      [3]   BIT STRING }

Versions ::= BIT STRING (v1988(0))

DirectoryBindResult   ::= DirectoryBindArgument
```

```
DirectoryBindError ::= SET {
      versions [0] Versions DEFAULT v1988,
      CHOICE {
         serviceError    [1]   ServiceProblem,
         securityError   [2]   SecurityProblem }}

DirectoryUnbind ::= ABSTRACT-UNBIND
      FROM {readPort, searchPort, modifyPort }

-- operations, arguments, and results --

Read ::= ABSTRACT-OPERATION
      ARGUMENT  ReadArgument
      RESULT    ReadResult
      ERRORS {
            AttributeError, NameError,
            ServiceError, Referral, Abandoned,
            SecurityError }

ReadArgument ::= OPTIONALLY-SIGNED SET {
      object        [0] Name,
      selection     [1] EntryInformationSelection
                              DEFAULT {},
      COMPONENTS OF CommonArguments }

ReadResult ::= OPTIONALLY-SIGNED SET {
      entry    [0] EntryInformation,
      COMPONENTS OF CommonResults }

Compare ::= ABSTRACT-OPERATION
      ARGUMENT  CompareArgument
      RESULT    CompareResult
      ERRORS {
            AttributeError, NameError,
            ServiceError, Referral, Abandoned,
            SecurityError }

CompareArgument ::= OPTIONALLY-SIGNED SET {
      object        [0] Name,
      purported     [1] AttributeValueAssertion,
      COMPONENTS OF CommonArguments }

CompareResult ::= OPTIONALLY-SIGNED SET {
      DistinguishedName OPTIONAL,
      matched   [0] BOOLEAN,
      fromEntry [1] BOOLEAN DEFAULT TRUE,
      COMPONENTS OF CommonResults }

Abandon ::= ABSTRACT-OPERATION
      ARGUMENT AbandonArgument
      RESULT AbandonResult
      ERRORS {AbandonFailed}

AbandonArgument ::= SEQUENCE {
      InvokeID [0] InvokeID}

AbandonResult ::=  NULL

List ::= ABSTRACT-OPERATION
      ARGUMENT  ListArgument
      RESULT    ListResult
      ERRORS {
            AttributeError, NameError,
            ServiceError, Referral, Abandoned,
            SecurityError }

ListArgument ::= OPTIONALLY-SIGNED SET {
      object [0] Name,
      COMPONENTS OF CommonArguments }
```

```
ListResult      ::= OPTIONALLY-SIGNED CHOICE{
        listInfo SET {
        DistinguishedName OPTIONAL
        subordinates [1] SET OF SEQUENCE {
                RelativeDistinguishedName,
                aliasEntry    [0]  BOOLEAN DEFAULT FALSE,
                fromEntry     [1]  BOOLEAN DEFAULT TRUE },
                partialOutcomeQualifier [2] PartialOutcomeQualifier
                                            OPTIONAL,
                COMPONENTS OF CommonResults},
        uncorrelatedListInfo [0] SET OF
                ListResult }

PartialOutcomeQualifier ::= SET {
        limitProblem    [0]   LimitProblem OPTIONAL,
        unexplored      [1]   SET OF
                ContinuationReference OPTIONAL,
        unavailableCriticalExtensions [2] BOOLEAN DEFAULT FALSE }

LimitProblem ::= INTEGER {
        timeLimitExceeded (0),
        sizeLimitExceeded  (1),
        administrativeLimitExceeded (2) }

Search ::= ABSTRACT-OPERATION
        ARGUMENT  SearchArgument
        RESULT    SearchResult
        ERRORS {
                AttributeError, NameError,
                ServiceError, Referral, Abandoned,
                SecurityError }

SearchArgument ::=  OPTIONALLY-SIGNED SET {
        baseObject      [0]  Name,
        subset          [1]  INTEGER {
                baseObject(0),
                oneLevel(1),
                wholeSubtree(2)} DEFAULT baseObject,
        filter          [2]  Filter DEFAULT and {},
        searchAliases   [3]  BOOLEAN DEFAULT TRUE,
        selection       [4]  EntryInformationSelection DEFAULT {},
        COMPONENTS OF CommonArguments }

SearchResult ::=        OPTIONALLY-SIGNED
        CHOICE {
        searchInfo SET {
        DistinguishedName OPTIONAL,
        entries  [0]  SET OF EntryInformation,
        partialOutcomeQualifier
           [2] partialOutcomeQualifier OPTIONAL,
        COMPONENTS OF CommonResults },
        uncorrelatedSearchInfo [0] SET OF
                SearchResult }

AddEntry ::= ABSTRACT-OPERATION
        ARGUMENT  AddEntryArgument
        RESULT    AddEntryResult
        ERRORS {
                AttributeError, NameError,
                ServiceError, Referral, SecurityError
                UpdateError }

AddEntryArgument ::= OPTIONALLY-SIGNED SET {
                object   [0]  DistinguishedName,
                entry    [1]  SET OF Attribute,
                COMPONENTS OF CommonArguments}

AddEntryResult ::=  NULL
```

```
RemoveEntry ::= ABSTRACT-OPERATION
      ARGUMENT RemoveEntryArgument
      RESULT   RemoveEntryResult
      ERRORS {
            NameError,
            ServiceError, Referral, SecurityError,
            UpdateError}

RemoveEntryArgument ::= OPTIONALLY-SIGNED SET {
            object  [0] DistinguishedName,
            COMPONENTS OF CommonArguments }

RemoveEntryResult ::= NULL

ModifyEntry ::= ABSTRACT-OPERATION
      ARGUMENT ModifyEntryArgument
      RESULT   ModifyEntryResult
      ERRORS {
            AttributeError, NameError,
            ServiceError, Referral, SecurityError,
            UpdateError}

ModifyEntryArgument ::= OPTIONALLY-SIGNED SET {
      object    [0]  DistinguishedName,
      changes   [1]  SEQUENCE OF EntryModification,
      COMPONENTS OF CommonArguments }

ModifyEntryResult ::= NULL

EntryModification ::= CHOICE {
      addAttribute    [0]  Attribute,
      removeAttribute [1]  AttributeType,
      addValues       [2]  Attribute,
      removeValues    [3]  Attribute}

ModifyRDN ::= ABSTRACT-OPERATION
      ARGUMENT ModifyRDNArgument
      RESULT   ModifyRDNResult
      ERRORS {
            NameError,
            ServiceError, Referral, SecurityError,
            UpdateError }

ModifyRDNArgument ::= OPTIONALLY-SIGNED SET {
      object        [0] DistinguishedName,
      newRDN        [1] RelativeDistinguishedName,
      deleteoldRDN  [2] BOOLEAN DEFAULT FALSE,
      COMPONENTS OF CommonArguments }

ModifyRDNResult ::= NULL

-- errors and parameters --

Abandoned ::= ABSTRACT-ERROR  -- not literally an "error"

AbandonFailed ::= ABSTRACT-ERROR
      PARAMETER SET {
         problem   [0] AbandonProblem,
         operation [1] InvokeID}

AbandonProblem ::= INTEGER {
          noSuchOperation (1),
          tooLate (2),
          cannotAbandon (3)}
```

```
AttributeError ::=    ABSTRACT-ERROR
      PARAMETER SET {
         object      [0]  Name,
         problems    [1]  SET OF SEQUENCE {
            problem        [0]  AttributeProblem,
            type           [1]  AttributeType,
            value          [2]  AttributeValue OPTIONAL }}

AttributeProblem ::=
      INTEGER {
           noSuchAttributeOrValue (1),
           invalidAttributeSyntax (2),
           undefinedAttributeType (3),
           inappropriateMatching (4),
           constraintViolation (5),
        attributeOrValueAlreadyExists (6) }

NameError ::= ABSTRACT-ERROR
      PARAMETER SET {
            problem [0]  NameProblem,
            matched [1]  Name}

NameProblem ::=    INTEGER {
            noSuchObject (1),
            aliasProblem (2),
            invalidAttributeSyntax (3),
            aliasDereferencingProblem (4)}

Referral ::= ABSTRACT-ERROR  -- not literally an "error"
      PARAMETER SET {
            candidate [0]  ContinuationReference}

SecurityError ::=    ABSTRACT-ERROR
      PARAMETER SET {
         problem [0] SecurityProblem }

SecurityProblem ::= INTEGER {
      inappropriateAuthentication (1),
      invalidCredentials (2),
      insufficientAccessRights (3),
      invalidSignature (4),
      protectionRequired (5),
      noInformation (6) }

ServiceError ::= ABSTRACT-ERROR
      PARAMETER SET {
           problem [0] ServiceProblem }

ServiceProblem ::=   INTEGER {
      busy (1),
      unavailable (2),
      unwillingToPerform (3),
      chainingRequired (4),
      unableToProceed (5),
      invalidReference (6),
      timeLimitExceeded (7),
      administrativeLimitExceeded (8),
      loopDetected (9),
      unavailableCriticalExtension (10),
      outOfScope (11),
      ditError (12) }

UpdateError ::= ABSTRACT-ERROR
      PARAMETER SET {
            problem  [0]  UpdateProblem }
```

```
UpdateProblem ::= INTEGER {
        namingViolation (1),
        objectClassViolation (2),
        notAllowedOnNonLeaf (3),
        notAllowedOnRDN (4),
        entryAlreadyExists (5),
        affectsMultipleDSAs (6),
        objectClassModificationProhibited (7)}

-- common arguments/results --

CommonArguments ::= SET {
        [30] ServiceControls DEFAULT {}
        [29] SecurityParameters DEFAULT {},
        requestor [28] DistinguishedName OPTIONAL,
        [27] OperationProgress DEFAULT notStarted,
        aliasedRDNs [26] INTEGER OPTIONAL,
        extensions [25] SET OF Extension OPTIONAL }

Extension ::= SET {
        identifier      [0] INTEGER,
        critical        [1] BOOLEAN DEFAULT FALSE,
        item            [2] ANY DEFINED BY identifier }

CommonResults ::= SET {
        [30] SecurityParameters OPTIONAL,
        performer [29] DistinguishedName OPTIONAL,
        aliasDereferenced [28] BOOLEAN DEFAULT FALSE}

-- common data types --

ServiceControls ::= SET {
        options [0]     BIT STRING {
        preferChaining (0),
        chainingProhibited (1),
        localScope (2),
        dontUseCopy (3),
        dontDereferenceAliases(4)}
                DEFAULT{},

        priority [1] INTEGER {
            low (0),
            medium (1),
            high (2) } DEFAULT medium,

timeLimit   [2]     INTEGER OPTIONAL,

sizeLimit   [3]     INTEGER OPTIONAL,

scopeOfReferral [4] INTEGER {
                dmd(0),
                country(1)}
                OPTIONAL }

EntryInformationSelection ::= SET {
        attributeTypes
            CHOICE {
                allAttributes [0] NULL,
                select [1] SET OF AttributeType
                -- empty set implies no attributes
                -- are requested --}
            DEFAULT allAttributes NULL,

        infoTypes [2] INTEGER {
                attributeTypesOnly (0),
                attributeTypesAndValues (1) } DEFAULT
                        attributeTypesandValues }
```

```
EntryInformation ::= SEQUENCE {
        DistinguishedName,
        fromEntry BOOLEAN DEFAULT TRUE,
        SET OF CHOICE {
                AttributeType,
                Attribute} OPTIONAL }

Filter ::=      CHOICE {
        item    [0]     FilterItem,
        and     [1]     SET OF Filter,
        or      [2]     SET OF Filter,
        not     [3]     Filter }

FilterItem      ::= CHOICE {
        equality        [0] AttributeValueAssertion,
        substrings      [1] SEQUENCE {
            type AttributeType,
            strings     SEQUENCE OF CHOICE {
                initial     [0]     AttributeValue,
                any         [1]     AttributeValue,
                final       [2]     AttributeValue}},

        greaterOrEqual [2]      AttributeValueAssertion,
            lessOrEqual         [3]     AttributeValueAssertion,
            present             [4]     AttributeType,
        approximateMatch        [5]     AttributeValueAssertion }

SecurityParameters ::= SET {
        certification-Path  [0] CertificationPath OPTIONAL,
        name        [1]     DistinguishedName OPTIONAL,
        time        [2]     UTCTime OPTIONAL,
        random      [3]     BIT STRING OPTIONAL,
        target      [4]     ProtectionRequest OPTIONAL }

ProtectionRequest ::= INTEGER {
                        none(0),
                        signed (1)}
```

附 件 B

（附于建议 X.511）

号码簿客体标识符

本附件是本标准的一部分。

本附件包括所有在本建议中使用的 ASN.1 客体标识符，并以 ASN.1 模块 **DirectoryObjectIdentifiers** 的形式描述。

```
DirectoryObjectIdentifiers {joint-ISO-CCITT ds(5) modules(1)
                            directoryObjectIdentifiers(9)}
DEFINITIONS ::=
BEGIN

EXPORTS
    id-ot-directory, id-ot-dua, id-pt-read, id-pt-search, id-pt-modify;

IMPORTS
    id-ot, id-pt
  FROM UsefulDefinitions {joint-iso-ccitt ds(5) modules(1),
                          usefulDefinitions(0)};
```

```
-- Objects --
id-ot-directory     OBJECT IDENTIFIER ::= {id-ot 1}
id-ot-dua           OBJECT IDENTIFIER ::= {id-ot 2}
-- Port Types --
id-pt-read          OBJECT IDENTIFIER ::= {id-pt 1}
id-pt-search        OBJECT IDENTIFIER ::= {id-pt 2}
id-pt-modify        OBJECT IDENTIFIER ::= {id-pt 3}
END
```

建　议　X.518

号码簿分布式操作规程[①]

（1988年订于墨尔本）

目　录

第1部分 — 绪论 …… 118

0　绪论 …… 118
1　应用范围 …… 118
2　参考文献 …… 118

3　定义 …… 119
4　缩写词 …… 120
5　符号法 …… 120

第2部分 — 概述 …… 120

6　概述 …… 120

第3部分 — 分布式号码簿模型 …… 121

7　分布式号码簿系统模型 …… 121
8　DSA交互操作模式 …… 122
　8.1　连锁转移 …… 122
　8.2　多点传播 …… 123
　8.3　转移参考 …… 124
　8.4　方式的确定 …… 125
9　号码簿分布 …… 125

① 建议X.518和ISO.9594－4，信息处理系统—开放系统互连—号码簿分布式操作规程是在紧密合作下制定的，而且在技术上是协调的。

10 知识 …… 127
10.1 最低限度的知识参考 …… 127
10.2 根上下文 …… 127
10.3 知识参考 …… 128
10.4 知识的管理 …… 129

第 4 部分 — *DSA* 抽象服务 …… 130

11 *DSA* 抽象服务概述 …… 130
12 信息类型 …… 131
12.1 绪论 …… 131
12.2 在其他处定义的信息类型 …… 131
12.3 连锁自变量 …… 132
12.4 连锁结果 …… 133
12.5 操作进展 …… 134
12.6 跟踪信息 …… 134
12.7 参考类型 …… 134
12.8 访问点 …… 135
12.9 连续参考 …… 135
13 抽象耦合与抽象释放 …… 135
13.1 DSA 耦合 …… 135
13.2 DSA 释放 …… 136
14 连锁抽象操作 …… 136
15 连锁抽象差错 …… 137
15.1 绪论 …… 137
15.2 DSA 转移参考 …… 138

第 5 部分 — 分布式操作规程 …… 138

16 绪论 …… 138
16.1 范围与界限 …… 138
16.2 概念模型 …… 138
16.3 DSA 的个别操作与集体操作 …… 138
17 分布式号码簿的性能 …… 139
17.1 通过协作完成的操作 …… 139
17.2 操作处理阶段 …… 139
17.3 分布式操作的管理 …… 140
17.4 有关分布式操作的其他考虑 …… 141
17.5 分布式操作的验证 …… 142
18 DSA 的特性 …… 143
18.1 绪论 …… 143
18.2 DSA 特性概述 …… 143
18.3 特定操作 …… 145

18.4 操作调度…… 146
18.5 循环操作…… 149
18.6 名称鉴别规程…… 150
18.7 客体评价规程…… 157
18.8 结果合并规程…… 159
18.9 分布式验证规程…… 160

附件 A — 分布式操作使用的 ASN.1 …… 161
附件 B — 知识模型 …… 165
附件 C — 验证的分布式应用 …… 169
附件 D — 分布式号码簿客体标识符 …… 173

第 1 部分 — 绪论

0 绪论

0.1 本文和本系列其他文件的目的在于简化信息处理系统之间的互连，以便提供号码簿服务。全套系统，包括它们的号码簿信息是一个整体，叫做号码簿。号码簿包含的全部信息叫做号码簿信息库(DIB)，其典型用途是简化客体之间以及同这些客体，或围绕这些客体(如应用实体、人员、终端和分布表)的通信联系。

0.2 号码簿在开放系统互连中扮演重要角色；开放系统的目的是除使用互连标准之外，设法只通过很少的技术协议就能实现下列信息处理系统之间的互连：

— 不同厂家；
— 不同管理部门；
— 不同复杂程度；
— 不同年代。

0.3 本建议为号码簿的分布式部件提供互连规程，使它们能够向用户提供协调的服务。

1 应用范围

1.1 本建议说明参与分布式号码簿操作的 DSA 的性能。DSA 的性能设计要求是：DIB 在多个 DSA 之间，在分布很广 的环境下保证为用户提供协调的服务。

1.2 号码簿的目的不是为了提供通用数据库系统(虽然可以在这样的系统上建立号码簿)。建立号码簿的假设条件是：查询服务的频率比修改号码簿的频率高得多。

2 参考文献

建议 X.200 — 开放系统互连基本参考模型
建议 X.208 — 开放系统互连抽象语法符号第 1 集(ASN.1)的规范
建议 X.500 — 号码簿概念、模型和服务概述
建议 X.501 — 号码簿模型
建议 X.511 — 号码簿抽象服务定义
建议 X.519 — 号码簿协议描述
建议 X.520 — 号码簿属性类型集
建议 X.521 — 号码簿客体类别集

3 定义

本节的定义使用 § 4 定义的缩写词。

3.1 *OSI* 参考模型定义

本建议使用建议 *X*.200 定义的术语。

a) *application entity title* 应用实体名称。

3.2 基本号码簿定义

本建议使用建议 *X*.500 定义的下列术语：

a) (*the*) *Directory* 号码簿；

b) *Directory Information Base* 号码簿信息库。

3.3 号码簿模型定义

本建议使用建议 X.501 定义的下列术语：

a) *access point* 访问点；

b) *alias* 别名；

c) *distinguished name* 可识别名；

d) *Directory Information Tree* 号码簿信息树；

e) *Directory System Agent* 号码簿系统代理；

f) *Directory User Agent* 号码簿用户代理；

g) *relative distinguished name* 相对可识别名。

3.4 抽象服务定义规范

本建议使用建议 *X*.407 定义的下列术语：

a) *abstract error* 抽象差错；

b) *abstract operation* 抽象操作；

c) *result* 结果。

3.5 分布式操作定义

本建议使用下面定义的术语：

a) *chaining* 连锁转移：自身不能完成某项操作的 DSA 可选择使用这种交互式操作方式。DSA 的连锁操作如下：DSA 求助于其他 DSA 的操作，然后把结果送到原请求者；

b) *context prefix* 上下文前缀：由 DIT 的根开始至命名上下文的初始极点为止的 RDN 序列，它相当于该极点的可识别名；

c) cross reference 交叉参考：一种知识参考，它包含有关保存登录项的 DSA 的情报。本参考使用于优化条件。登录项不需要有上下级关系。

d) *DIB fragment* *DIB* 片断：它是 DSA 保存的 DIB 的一部分：它包含一个或多个命名上下文；

e) *distributed name resolution* 分布式名称鉴别：在多个 DSA 中进行的名称鉴别过程；

f) *internal reference* 内部参考：知识参考，它包含指向同一个 DSA 的某一个登录项的内部指针；

g) *knowledge information* 知识信息：特定 DSA 掌握的有关所属登录项的情报，以及有关如何从号码簿中寻找其他登录项的情报；

h) *knowledge reference* 知识参考：用于（直接地或不直接地）联系 DIT 登录项和它所在的 DSA 的知识；

i) *knowledge tree* 知识树：DSA 掌握的知识信息的概念化模型，DSA 依靠它对分布式名称进行鉴别；

j) *multicasting* 多点传播：交互式操作的一种方式，可供自身不能完成某项操作的 DSA 任意选用。DSA 将操作多点传播，即 DSA 求助于其他的 DSA 共同完成同一个操作（串联或并联），并且把适当的结果送回原请求者；

k) *name resolution* 名称鉴别：把声称名的每一个 RDN 同 DIT 的极点对照起来寻找一个合适的登录项的过程；

l) *naming context* 命名上下文：DIT 的部分子树，从极点开始往下延伸到叶和/或非叶极点止。这些极点构成命名上下文的边界。属于该边界的非叶极点表示其他命名上下文的起始点；

m) *non－specific subordinate reference* 非特定下级参考：一种知识参考，它包含有关特定 DSA 的情报，该 DSA 拥有一个或多个未指定的下级登录项；

n) *operation progress* 操作进展：一组值，它指示名称鉴别的进展情况；

o) *reference path* 参考路径：一个连续的知识参考序列；

p) *referral* 转移参考：由自身不能完成操作的 DSA 回送的一种结果，它指定一个或多个能够完成操作的其他 DSA；

q) *request decomposition* 请求分解：把一个请求分解为若干子请求，每一个子请求负责完成分布式操作的一部分；

r) *root context* 根上下文：极点的命名上下文，该极点的名称包含各 RDN 的空序列；

s) *subordinate reference* 下级参考：一种知识参考，它包含有关拥有特定下级登录项的 DSA 的情报；

t) *subrequest* 子请求：对请求进行分解而产生的一个请求；

u) *superior reference* 超级参考：一种知识参考，它包含有关拥有超级登录项的 DSA 的情报。

4 缩写词

本建议采用下列缩写词：

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

RDN 相对可识别名

5 符号法

本节使用的符号定义如下：

a) 数据语法符号、编码与宏符号在建议 X.208 中定义；

b) 抽象模型与抽象服务的符号在建议 X.407 中定义。

第 2 部分 — 概述

6 概述

号码簿抽象服务允许用户对 DIB 中的号码簿信息进行查询、检索和修改。此种号码簿抽象服务，在建议 X.511 中是用抽象号码簿客体表示的。

当然，抽象号码簿客体的规范丝毫未涉及到号码簿的物理实现，即未涉及到 DSA 的规范。DSA 负责储存、管理 DIB 和提供号码簿服务。该规范不考虑 DIB 是否集中(即存放在一个 DSA)或分布于多个 DSA 之中。因此，服务说明不包括为了在一个分布式环境中支持抽象服务对 DSA 提出的一系列要求，如要求了解其他 DSA 的情况、转移到其他 DSA，和同其他 DSA 进行合作。

本建议描述抽象号码簿客体的详细概念，并以构成分布式号码簿服务的一个或多个 DSA 客体表示。其中，最基本的是对号码簿客体内部的 DSA 端口进行识别和说明。本建议还描述每一个端口的相关抽象服务及端口规程。

此外，本建议还描述 DIB 在一个或多个 DSA 之中分布的适当方式。当 DIB 只存在于一个 DSA 时，号码簿实际上是集中式；当 DIB 分布于两个或多个 DSA 时，为了保证所有包含特定登录项的 DSA 都能够访问整个 DIB，本建议规定了“知识”和转移机制。

另外，本建议还描述了专门用来处理用户请求的一些交互式操作方式，便于用户对号码簿的某些操作特性进行控制。具体地说，用户对响应查询请求的 DSA 能够进行这样的控制：当用户需要查询储存于其他 DSA 的信息时，响应请求的 DSA 可选择直接(连锁转移或多点传播)向其他的 DSA 查询，或在响应中提供有关能够把查询请求(转移参考)继续往前传递的 DSA 的情报。

在一般情况下，DSA 是否要进行连锁转移、多点传播操作或委托其他的 DSA，取决于用户设置的服务控制，以及 DSA 自己的行政、营业或技术情况。

考虑到一般情况，号码簿为分布式，为了提供号码簿查询服务需要有多个 DSA 遵照上述准则进行连锁转移或多点传播操作或委托另一个 DSA 协助完成任务，本建议特地为 DSA 提供响应分布式号码簿查询时所需要的适当规程。这些规程能保证分布式号码簿为用户提供友善和协调的服务。

第 3 部分 — 分布式号码簿模型

7 分布式号码簿系统模型

在建议 X.511 中定义的号码簿抽象服务，利用为用户提供一套号码簿服务的客体塑造号码簿模型。号码簿服务模型是以端口表示的，每一个端口提供一套特定号码簿服务。用户通过访问点接受号码簿服务。号码簿允许有一个或多个访问点；每一个访问点根据所提供的服务性质和为提供这些服务使用的交互式操作方式进行分类。

本节讨论号码簿客体的内部结构和构造号码簿的客体及其端口，目的是为了简化对分布式号码簿服务的描述。

图 1/X.518 表示分布式号码簿。根据这个示意图可很方便地描述号码簿的分布特性。该图表示了由一个或多个 DSA 客体组成的号码簿客体。

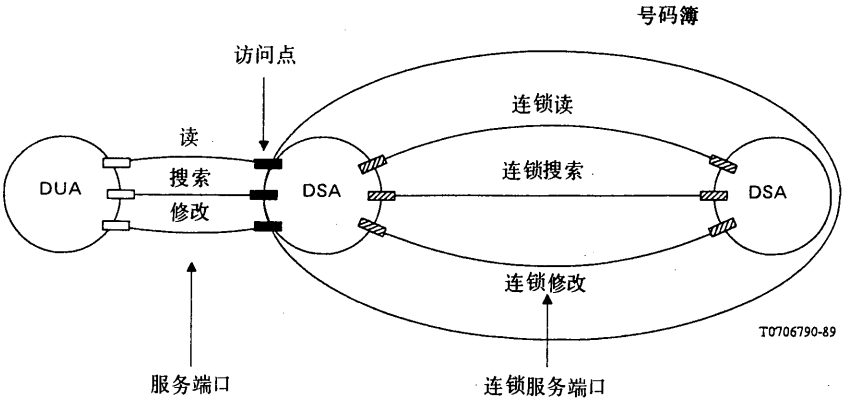


图 1/X.518

分布式号码簿模型的客体

DSA 客体的详细描述将在本建议的其余条款下给出。本条款 只列举 DSA 客体的一些特性，做为一般性介绍，并指出本建议与其他建议之间的关系。

定义 DSA 客体时应考虑到 DIB 的分布，为了向号码簿用户(DUA)提供号码簿服务，要求在物理上分散布局的 DSA 能够按照预定的合作方式进行联系。

DSA 客体的特性和号码簿客体均使用从外部看得见的端口来表示。与 DSA 客体有关联的端口分为两种类型：服务端口和连锁服务端口。

DSA 客体的服务端口和号码簿客体端口是一样的，即分为**读**、**搜索**和**修改**端口。图 1/X.518 说明与 DSA 客体有关联的服务端口是一个访问点，通过访问点提供号码簿服务。

有关**读**、**搜索**和**修改**服务端口的详细说明在建议 X.511 中描述(根据这些端口定义制定的相应的 OSI 应用服务要素协议规范在建议 X.519 中描述)。

除了 DSA 客体服务端口之外，还定义了第二套端口，即连锁服务端口。这些端口是为了 DSA 之间的通信而设置的，以便在分布式环境中实现号码簿抽象服务。

连锁服务端口和通过这些端口提供的操作的名称与服务端口的类似名称是对称的，即它们分别叫做连锁读(**chainedRead**)、连锁搜索(**chainedSearch**)和连锁修改(**ChainedModify**)。

对更抽象的客体的成员客体的描述过程称为“细节性”描述。本建议第 4 部分给出对号码簿客体分解为组成部分(DSA)的细节性描述以及对各组成部分提供的抽象服务(DSA 抽象服务)的描述。建议 X.519 给出根据连锁端口定义制定的相应的 OSI 应用服务要素协议规范。

8 DSA 交互操作模式

号码簿的基本特性描述如下：给出一个分布式 DIB，用户就能够通过任何访问点获得所请求的任何服务(但必须符合保密、访问控制以及行政政策)。为了满足上述要求，任何 DSA 在提供某种服务之前必须知道用户要寻找的信息存放的地点(见本建议 § 10)，然后负责将这个情报送到请求者手里，或自己设法满足用户的请求(请求者可能是 DUA 或另一个 DSA：在后一种情况下，两个 DSA 都必须拥有连锁端口)。

为了满足上述要求，定义了三种 DSA 交互式操作方式：“连锁”、“多点传播”和“转移参考”方式。“连锁”和“多点传播”方式适用于请求者是另一个 DSA 的场合；“转移参考”方式则适用于请求者为一个 DUA 的场合。

8.1 连锁转移

这种交互式操作方式(如图 2/X.518 所示)可用于 DSA 向另一个 DSA 传递请求的场合，其条件是前者必须知道掌握在后者手里的命名上下文。连锁方式还可用于联系交叉参考、下级参考或上级参考中指出的 DSA。多点传播方式是连锁方式的一种，它在 § 8.2 中描述。

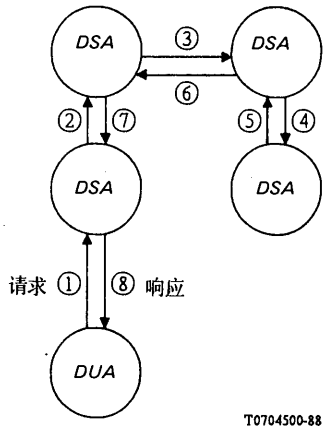


图 2/X.518

连 锁 方 式

注 — 图 2/X.518 中所示的相互作用线及其编号表示交互式操作过程的顺序。

8.2 多点传播

这种交互式操作方式(如图 3a/X.518 和 36/X.518 所示)适用于在某一个 DSA 不知道存放于其他 DSA 的全部命名上下文的情况下需要把同一个请求以并联(a)或串联(b)方式向另一个或多个 DSA 连锁传递的场合。仅当 DSA 需要同非特定下级参考中指出的其他 DSA 进行联系时才使用多点传播方式。每一个 DSA 都收到同样的请求。在一般情况下,在名称鉴别过程中只有其中的一个 DSA 能够继续处理远程操作,其他的 DSA 则回送 **unableToProceed ServiceError** 信息。但是,在搜索和列表操作的评价阶段,非特定下级参考中的所有 DSA 都应当能够继续对请求进行处理。

注 — 图 3a/X.518 和 3b/X.518 中所示的相互作用线及其编号表示交互式操作过程中的顺序。

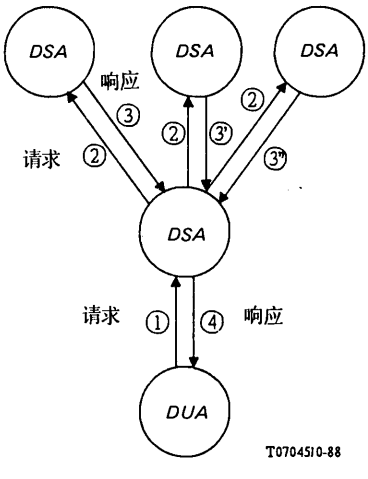


图 3a/X.518

多点传播方式

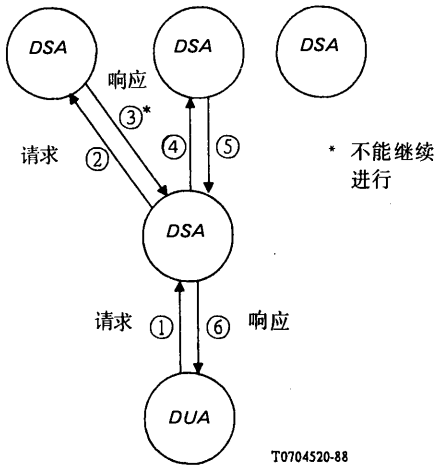


图 3b/X.518

多点传播方式

8.3 转移参考

当有一个 DUA 或另一个 DSA(在此种情况下,两个 DSA 都必须拥有一个连锁服务端口)向某一个 DSA 请求完成某一项请求时,该 DSA 就在它的响应中回送一个转移参考(如图 4a/X.518 和 4b/X.518 所示)。该转移参考可以是全部响应(在此情况下,它应被列为一个差错),或者是响应的一部分。转移参考包含一个知识参考,后者可以是上级参考、下级参考、交叉参考或者是非特定下级参考。

接受转移参考的 DSA(图 4a/X.518)可利用其中的知识参考将原操作传播给或连锁转移给其他的 DSA(视参考类型而定)。另外,收到转移参考的 DSA 也可以在它的响应中回送该转移参考。收到转移参考的 DUA(图 4b/X.518)可利用该转移参考联系一个或多个 DSA 来向前推进该请求。

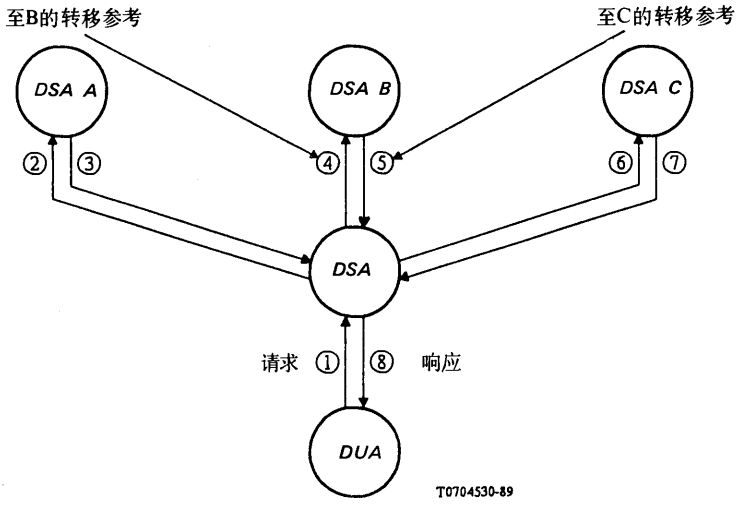


图 4a/X.518

转移参考方式—DSA 使用连锁端口

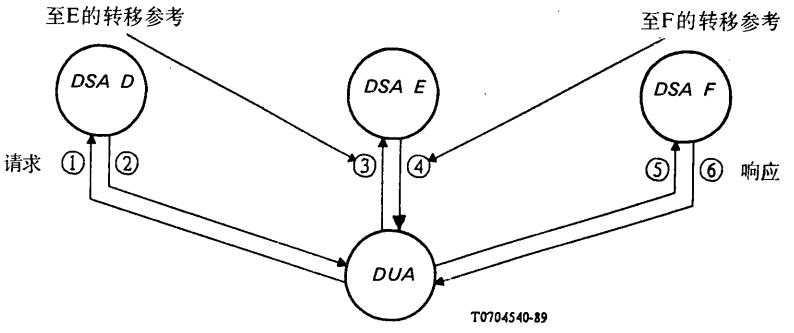


图 4b/X.518

转移参考—DUA 不使用连锁端口向 DSA 发送请求

注 — 图 4a/X.518 和 4b/X.518 中的相互作用线及其编号表示交互式操作过程的顺序。

8.4 *方式的确定*

倘若 DSA 本身不能满足某一个请求，则它必须将该请求(或经过分解之后的原请求)向其他 DSA 连锁转移或传播，但在下列情况下除外：

a) 用户通过服务控制禁止使用连锁方式进行交互式操作；在此情况下 DSA 必须回送一个转移参考或 **chainingRequiredServiceError**(任选)，或者

b) DSA 因行政、操作或技术上的原因决定不采用连锁方式；在此种情况下 DSA 必须回送一转移参考。

注 *1* — 不采用连锁或多点传播方式的技术原因，是在知识参考中指出的 DSA 没有连锁服务端口。

注 *2* — 倘若 **localScope** 服务控制被置位，则 DSA(或 DMD)必须满足请求或回送一个差错信息。

注 *3* — 倘若用户选择使用转移参考，则该用户必须将 **chainingProhibited** 置位。

9 号码簿分布

本节定义分布式 DIB 的准则。

DIB 内的每一个登录项只属于一个、而且是唯一的(DSA 的)主管部门，后者对该登录项享有行政管辖权。登录项的维护和管理必须在一个由管理该登录项的行政权威所管辖的 DSA 中实现。

尽管号码簿不提供复制登录项的手段，但它仍然能够复制登录项，方法有以下两种：

— 通过双边协商把登录项的拷贝储存在其他的 DSA 内。管理和维护这些拷贝的手段必须通过双边协商确定，本建议不定义这种手段。

— 登录项的拷贝可通过在本地和动态地储存登录项的拷贝(通过请求)获得。

注 — 储存的登录项可通过访问控制获得。

发送请求的用户应当收到(**fromCopy**)通知,说明他收到的信息是否为登录项的一个拷贝。本建议定义了一个服务控制 **dontUseCopy**,使用户能够要求禁止使用登录项的拷贝。

在号码簿里的每一个 DSA 都保存 DIB 的片断。DSA 保存的 DIB 片断是用 DIT 描述的,它包含一个或多个命名上下文。所谓*命名上下文*是 DIT 的部分子树,它从一个极点开始向下延伸到叶和/或非叶极点。这样的极点构成命名上下文的边界。属于边界的非叶极点的下级表示下一层命名上下文的开端。

DSA 的管理部门有可能对一些分离的命名上下文拥有行政管理权。受 DSA 的管理部门管辖的每一个命名上下文在逻辑上必须包含从 DIT 的根起到包含该命名上下文的子树的初始极点止的 RDN 序列。这个 RDN 序列叫做*上下文前缀*。

DSA 可以把管理在本地储存的任何登录项的任何直接下级的行政管理权授给别的 DSA。授权的 DSA 叫做上级 DSA,而包含被授权的 DSA 的上级登录项的上下文则叫做*上级命名上下文*。行政管理权的授权是从 DIT 的根开始向下进行的;即授权只能从某一个登录项到它的下级进行。

图 5/X.518 表示一个假想的 DIT,它在逻辑上分为五个命名上下文(A、B、C、D、E),各上下文在物理上分布在三个 DSA(DSA1,DSA2,DSA3)之中。

从实例中可以看出,特定 DSA 拥有的命名上下文可根据许多操作上的不同要求进行布置。通过设计可以使某些 DSA 包含特定登录项,该登录项在 DIB 的某些逻辑部分中代表高级命名域,例如一个大公司的组织结构,但不一定要包括所有的下级登录项。另一种选择是:经过设计可以使 DSA 只包含代表主要叶登录项的命名上下文。

从上述定义可知,一个命名上下文可以少至包含一个登录项,最多则可包含全部 DIT。

虽然 DIT 到 DSA 的逻辑到物理的映射可能是任意的,但若能使 DSA 只含少数命名上下文,则信息的布局和管理可以简化。

为了能够处理某一个请求,DUA 必须掌握一些信息,如至少掌握它所能接触到的第一个 DSA 的表示地址。至于 DUA 如何获得和储存这些信息则属于本地决策。

在修改登录项的过程中,有可能使号码簿前后不统一。若在不同的 DSA 修改别名或别名客体,则这个可能性尤其存在。上述不统一性必须通过行政措施予以纠正。例如,若别名客体已被删除,则也应删除相应的别名。在存在不统一的一段时间内,号码簿继续工作。

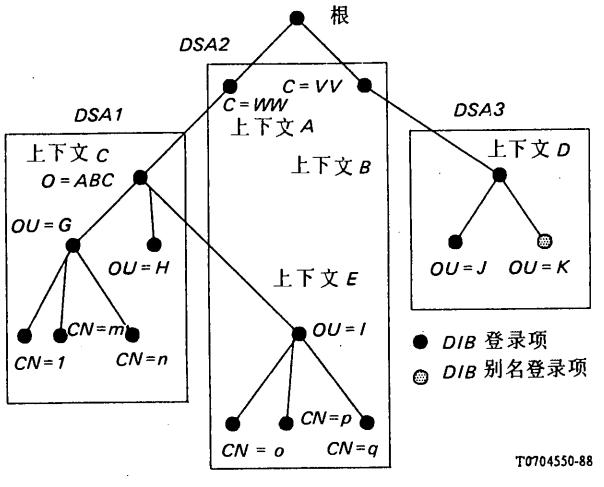


图 5/X.518

假想的 DIT

注 — DSA 不保存根，但在局部层次上应当提供某种标记，以识别代表根的直接下级的极点（例如：C=VV，C=WW）。

10 知识

DIB 有可能分布于多个 DSA 之中，每一个 DSA 均包含一个 DIB 片断。有关 DIB 的分布原则在本建议 §9 中描述。

对于号码簿，要求在用户使用某种交互式操作方式的情况下号码簿的分布保持透明，从而使整个 DIB 仿佛都存在于每一个 DSA 之中。

为了支持上述操作要求，包含 DIB 片断的每一个 DSA 都必须能够识别包含在其他 DSA 之中的其他 DIB 片断或（任选）能够同这些 DIB 片断进行交互式操作。

本节定义知识，作为将名称映射到该名称在 DIT 片断中的地址的基础。

从概念上讲，DSA 包含两种信息：

a） 号码簿信息；

b） 知识信息。

*号码簿信息*是包含命名上下文的一组登录项；特定 DSA 的管理部门对上述上下文享有行政管理权。

*知识信息*包含特定 DSA 所属的命名上下文，同时指出这些上下文在整个 DIT 等级中的地位。名称鉴别是根据知识信息操作的，它是根据给出的登录项名寻找 DSA 的过程，该 DSA 对上述登录项享有行政管理权。

*上下文前缀*是从 DIT 的根起延伸到命名上下文的初始极点止的 RDN 序列，而且相当于该极点的可识别名。

*命名上下文*包含一组知识参考和一个上下文前缀。一个命名上下文必须完全包含下列知识参考：

— 定义包含在命名上下文中的一部分 DIT 内部结构的所有内部参考。

— 指向别的命名上下文的所有下级参考和非特定下级参考。

10.1 最低限度的知识参考

号码簿的特点是，对每一个登录项都可独立进行访问，不管访问请求是在何处发起的。

为了达到上述目的，每一个 DSA 至少保持下列知识参考：

— *下级参考*（在 §10.3.2 中定义）和/或*非特定下级参考*（在 §10.3.5 中定义）；

— *上级参考*（在 §10.3.3 中定义）。

这样就有可能为号码簿内的所有命名上下文建立一条*参考路径*，即连续的知识参考序列。

为了优化操作，必要时可把 §10.3.4 中定义的*交叉参考*做为参考路径的一部分，但这是任选的。

10.2 根上下文

由于不同的国家或世界组织的自主性，可能没有一个 DSA 能够包含根上下文。涉及命名鉴别进程的“根－DSA”的功能，应当由对根的直接下级的命名上下文享有行政管理权的那些 DSA 负责提供。这些 DSA 叫做*一级 DSA*。每一个一级 DSA 必须能够模拟“根－DSA”的功能。为此需要具备有关根命名上下文的全部知识。每个一级 DSA 都有根上下文的拷贝，因此根上下文应由享有自主权的一级行政权威统一管理。行政管理规程应通过多边协商制定，它不属于本建议的讨论范围。

— 每个一级 DSA 必须包含根上下文，因此包含到达各个一级 DSA 的参考路径。

— 每个非一级 DSA 应包含一个上级参考，因此包含到达任何一级 DSA 的参考路径。

10.3 知识参考

DSA 掌握的知识是利用一个或多个知识参考表示的。每一个参考直接或间接地把 DIB 的登录项同包含这些登录项的 DSA 联系起来。

为了能够从任何 DSA 到达每一个 DIB 的登录项，每一个 DSA 必须掌握有关自己所保存的登录项的知识，以及有关下级，也可能是上级的知识。从而得出下列几种知识参考：

— 内部参考

— 下级参考

— 上级参考

— 非特定下级参考。

另外，为了优化目的，特此定义下列任选参考：

— 交叉参考

在与特定 DSA 相关的一组知识参考中只包含内部参考的情况下，DSA 不知道其他 DSA 的情况，故 DIB 是集中的。

10.3.1 内部参考

一个内部参考包含：

— DIB 登录项的相应 RDN；

— 指向登录项在本地 DIB 内储存的地址的指针。（有关指针的规范不属于本建议范围。）

特定 DSA 管辖的所有登录项是以该 DSA 的知识信息中的内部参考表示的。

10.3.2 下级参考

一个下级参考包含：

— 直接下级 DIB 登录项的相应 RDN；

— DSA 的访问点（管理该登录项的行政管理权已授给 DSA）。

其他 DSA 保存的所有下级登录项必须由下级参考（或 § 10.3.5 中所述的非特定参考）表示。上述 DSA 已由本 DSA 授以行政管理权。

10.3.3 上级参考

一个上级参考包含：

— DSA 的访问点。

每一个非一级 DSA 应保持一个上级参考。上级参考应构成指向根的部分参考路径。除非在本标准以外采用某种办法，例如在 DMD 内部，来保持上述要求，一般应求助于另一个 DSA；这个 DSA 的命名上下文的前缀包含的 RDN 数比上述非一级 DSA 的上下文前缀所包含的最小 RDN 数还要少。

倘若要引入一个新的非一级 DSA，则该 DSA 应当具备最低限度的初始知识，这种初始知识是由上级参考表示的。新知识将由下级参考或交叉参考（如 § 10.3.4 所述）负责补充。倘若引入新的一级 DSA，则它必须具备根上下文并通知其他一级 DSA。关于如何完成此功能，则不属于本建议的讨论范围。

10.3.4 交叉参考

一个交叉参考包含：

— 一个上下文前缀；

— DSA 的访问点，该 DSA 拥有管理命名上下文的行政管理权。

这种类型的参考是任选的，它们有利于对命名鉴别操作的优化。一个 DSA 可拥有任意个(包括零个)交叉参考。

10.3.5 非特定下级参考

一个非特定下级参考包括：

— DSA 的访问点，该 DSA 包含一个或多个直接下级命名上下文。

这种类型的参考是任选的，它涉及这样的情况：已知 DSA 包含一些下级登录项，但是它们的具体 RDN 却是未知的。

DSA 所包含的每一个命名上下文可包含任意个(包括零个)非特定下级参考，当找到了所有的特定内部参考和下级参考之后，才计算这些非特定下级参考的值。通过一个非特定参考访问的 DSA 必须能够直接解决某一个请求(不论失败或成功)。在宣告失败的情况下，应向请求者回送 **ServiceError**，报告 **unableToProceed** 问题。

10.4 知识的管理

为保证分布很广的号码簿在操作性能上和操作协调上达到令人满意的程度，需要制定一套规程来维护和扩展每一个 DSA 所包含的知识。这些规程对初始知识的建立也是适用的。

知识的维护如下：

a) 由 DSA 或其行政权威负责维护：当 DSA 或其行政权威修改它的知识时，所有与这个知识有关的旧参考都将失效，因此必须将有关知识的变化情况通知所有保存这些参考的 DSA。这是维护上、下级和非特定下级参考的唯一方法。

b) 由 DSA 负责维护：为了改善操作性能，该 DSA 利用普通号码簿操作去申请和获得交叉参考。

本建议未定义如 a)所述的知识变化扩散规程。为了实现上述目的，必须在本地进行双边协商。

10.4.1 申请交叉参考

为了改善号码簿系统的操作性能，可利用普通号码簿操作来扩展本地交叉参考。若 DSA 拥有一个连锁端口，则该 DSA 能够向其他的 DSA(它必须也拥有一个连锁端口)申请回送一些知识参考，这些参考包含与普通号码簿操作的目标客体名有关的命名上下文的地址信息。

倘若 **ChainingArgument** 中的 **returnCrossReference** 参数置为 **TRUE**，则 **ChainingResult** 中的 **crossReference** 参数可能存在，并且包含交叉参考项的序列。

倘若 DSA 无法把它的请求连锁转移到相邻的 DSA，则必须向发起者 DSA 回送一个转移参考。倘若连锁自变量中的 **returnCrossReference** 参数为 **TRUE**，则上述转移参考允许包含相关命名上下文的前缀。若转移参考是由非特定下级参考构成的，则 **contextPrefix** 参数不复存在。转移参考回送的交叉参考只根据生成该转移参考的 DSA 所掌握的知识构成。

在两种情况下(连锁操作结果和转移参考)，行政权威可通过它的 DSA 决定对回送交叉参考的请求不予

理睬。

10.4.2 知识的不一致性

为了确保知识的一致性，号码簿必须支持一致性检测机制。

10.4.2.1 知识不一致性检测

不同类型的知识参考产生不同的不一致性，故需要采用不同的检测方法。

— 交叉参考与下级参考：

倘若做为参考对象的 DSA 没有本地命名上下文，而且参考中不包含上下文前缀，则这种类型的参考是无效的。这种不一致性将在确定名称鉴别进程的初始命名上下文的过程中通过 **ChainingArgument** 的操作进展和参考类型成分所检测。

— 非特定下级参考：

倘若做为参考对象的 DSA 没有本地命名上下文(该上下文的直接上级上下文前缀包含在该非特定下级参考中，即非特定下级参考包含该 DSA 的本地上下文前缀减去最后的 RDN)，则这种类型的参考是无效的。一致性检测同上。

— 上级参考：

倘若上级参考不是指向根的参考路径的一部分，则这种上级参考是无效的。上级参考的维护必须通过外部手段实现，它不属于本建议的讨论范围。

注 — 不是在所有的情况下都能检测到无效的上级参考。

10.4.2.2 知识不一致性报告

倘若在完成号码簿请求的过程中采用了连锁方式，则保存无效知识参考的 DSA 通过接受包含 **invalidReference** 问题的 **ServiceError** 将检测到所有的知识不一致性。

倘若 DSA 回送一个包含无效知识参考的转移参考，又若申请者使用该转移参考，则该申请者将收到携带 **invalidReference** 问题的 **ServiceError**。有关如何将差错情况传播给储存无效参考的 DSA 的问题不属于本建议的讨论范围。

10.4.2.3 对不一致的知识参考的处理

一旦 DSA 检测到无效参考，它必须设法恢复知识的一致性。例如，通过删除无效交叉参考或通过使用 **requestCrossReference** 机制来获得正确参考，以取代无效参考。

DSA 对无效参考的处理方式属于本地决策，而不属于本建议的讨论范围。

第 4 部分—DSA 抽象服务

11 DSA 抽象服务概述

11.1 号码簿的抽象服务在建议 X.511 中描述。当在分布环境中提供号码簿抽象服务时(如本建议 § 7 给出的模型)，可设想抽象服务是由一组 DSA 共同提供的，如图 1/X.518 所示。

11.2 为了描述 DSA 抽象服务模型，可将 **directory** 客体分解为 **dsa** 客体成员如下：

```
DirectoryRefinement ::= REFINE directory AS
    dsa             RECURRING
        readPort                [S] VISIBLE
        searchPort              [S] VISIBLE
        modifyPort              [S] VISIBLE
        chainedReadPort         PAIRED with dsa
        chainedSearchPort       PAIRED with dsa
        chainedModifyPort       PAIRED with dsa
```

11.3 **dsa** 客体本身定义如下：

```
dsa OBJECT
    PORTS { readPort                [S],
            searchPort              [S],
            modifyPort              [S],
            chainedReadPort,
            chainedSearchPort,
            chainedModifyPort}
::= id-ot-dsa
```

DSA 提供读、搜索和修改端口，从而使这些服务呈显在号码簿客体的用户(即 DUA)眼前。此外，DSA 支持连锁端口，即连锁读，连锁搜索和连锁修改，从而使 DSA 能够把申请连锁服务的请求传播到其他 DSA。

11.4 在§11.2 和§11.3 描述的端口(在建议 X.511 中定义的除外)定义如下：

```
chainedReadPort     PORT
    ABSTRACT OPERATIONS {
        ChainedRead, ChainedCompare,
        ChainedAbandon}
    ::=     id-pt-chained-read
chainedSearchPort   PORT
    ABSTRACT OPERATIONS {
        ChainedList, ChainedSearch}
    ::=     id-pt-chained-search
chainedModifyPort   PORT
    ABSTRACT OPERATIONS {
        ChainedAddEntry,
        ChainedRemoveEntry,
        ChainedModifyEntry,
        ChainedModifyRDN}
    ::=     id-pt-chained-modify
```

12 信息类型

12.1 绪论

12.1.1 本章节列举和(在某些情况下)定义将在各种 DSA 抽象服务操作定义中使用的信息类型，即指那些对多种操作现在或将来都通用的，或十分复杂，或自成体系，以致有必要把它们和使用它们的操作分开定义的信息类型。

12.1.2 在 DSA 抽象服务的定义中使用的一些信息类型实际上已在别处定义。本文§12.2 列举这些类型并指出其定义的出处。其余章节(§12.3 至§12.9)分别列举并定义一种信息类型。

12.2 在其他处定义的信息类型

12.2.1 下列信息类型在建议 X.501 中定义：

a) **aliasedObjectName** 别名客体名；

b) **DistinguishedName** 可识别名；
c) **Name** 名称；
d) **RelativeDistinguishedName** 相对可识别名；

12.2.2 下列信息类型在建议 X.511 中定义：
(抽象耦合)；
a) **DirectoryBind** 号码簿耦合；
(抽象操作)
b) **Abandon** 放弃；
(抽象差错)
c) **Abandoned** 被放弃；
d) **AttributeError** 属性差错；
e) **NameError** 名称差错；
f) **SecurityError** 保密差错；
g) **ServiceError** 服务差错；
h) **UpdateError** 更新差错；
(宏)
i) **OPTIONALLY-SIGNED** 任选署名；
(数据类型)
j) **SecurityParameters** 保密参数。

12.2.3 下列信息类型在建议 X.520 中定义：
a) **PresentationAddress** 表示地址：

12.3 连锁自变量

12.3.1 **ChainingArguments** 出现在每一个连锁抽象操作之中，目的是为了向 DSA 传递为成功地完成整个任务中的一部分所需要的信息：

```
ChainingArguments ::=   SET {
    originator          [0]   DistinguishedName OPTIONAL,
    targetObject        [1]   DistinguishedName OPTIONAL,
    operationProgress   [2]   OperationProgress DEFAULT {notStarted},
    traceInformation    [3]   TraceInformation,
    aliasDereferenced   [4]   BOOLEAN DEFAULT FALSE,
    aliasedRDNs         [5]   INTEGER OPTIONAL,
        -- absent unless aliasDereferenced is TRUE
    returnCrossRefs     [6]   BOOLEAN DEFAULT FALSE,
    referenceType       [7]   ReferenceType DEFAULT superior,
    Info                [8]   DomainInfo OPTIONAL,
    timeLimit           [9]   UTCTime OPTIONAL,
                        [10]  SecurityParameters DEFAULT {}}
```

12.3.2 各自变量的含义在§12.3.2.1至§12.3.2.11中定义。

12.3.2.1 **originator** 表示(原)申请者的名称，除非该名称已在保密参数中指出。若 **requestor** 存在于 **CommonArguments** 之中，则此自变量可以省略。

12.3.2.2 自变量 **targetObject** 表示目标客体名，即该客体的号码簿登录项是操作目标。上述客体扮演的角色取决于相关的特定抽象操作：也许该客体的登录项是操作对象，或者该客体是包括多个客体(如 **ChainedList** 或 **ChainedSearch**)的请求、或次请求的基础客体。仅当此自变量的值同 **XArgument**(见§14.3.1)中的基础客体的参数值相等时才可以将此自变量省略，在此情况下，其隐含值就是该值。

12.3.2.3 自变量 **operatingProgress** 用于通知 DSA 关于操作的进展情况，从而告之 DSA 在总操作中应当完成的任务。此自变量传递的信息在 § 12.5 中定义。

12.3.2.4 自变量 **traceInformation** 在连锁操作过程中用于避免 DSA 与 DSA 之间产生循环。在 DSA 把操作传递给别的 DSA 之前应当先把新的元素加到跟踪信息中。当 DSA 收到欲完成某种操作的请求时，它必须检查跟踪信息，检查在操作过程中是否产生循环现象。此自变量表达的信息在 § 12.6 中定义。

12.3.2.5 自变量 **aliasedDereferenced** 是一个布尔值，它的作用是：指出是否曾经遇到过一个或多个别名登录项，在分布式名称鉴别过程中是否换过名。**FALSE** 的默认值表示没有别名登录项被换名。

12.3.2.6 自变量 **aliasedRDN** 指出在 **targetObjectName** 中的 RDN 有多少是从一个(或多个)别名登录项中的 **aliasedObjectName** 属性中生成的。每次遇到一个别名登录项并且换名就把整数值置位。仅当自变量 **aliasedDereferenced** 取 **TRUE** 值时，本自变量才需要出现。

12.3.2.7 自变量 **returnCrossRefs** 是一个布尔值，它指出：在完成分布式操作的过程中，是否需要把使用的知识参考连同一个结果或转移参考一起送回初始 DSA 做为交叉参考。默认值 **FALSE** 表示不要求回送知识参考。

12.3.2.8 自变量 **referenceType** 向被请求完成抽象操作的 DSA 指出使用的知识类型。因此，DSA 能够检测起动者保存的知识中的差错。倘若 DSA 检测到某一个差错，则它必须通过 **ServiceError** 指出 **invalidReference** 的问题。**ReferenceType** 在 § 12.7 中详述。

注 — 倘若找不到 **referenceType**，则应使用 **Superior** 值。

12.3.2.9 自变量 **info** 是用来向所有参与处理某一个请求的 DSA 传递特定 DMD 信息。此自变量属于 **DomainInfo** 类型，它是无限制类型：

```
DomainInfo      ::=  ANY
```

12.3.2.10 自变量 **timeLimit** 若出现，指出操作应当结束的时间。

12.3.2.11 自变量 **SecurityParameters** 在建议 X.511 中定义。当此自变量缺席时，应当认为保密参数的集合是空的。

12.4 连锁结果

12.4.1 **ChainingResults** 出现在每一个抽象操作结果中，并向申请该抽象操作的 DSA 提供反馈信息。

```
ChainingResults    ::=    SET {
      Info                [0]    DomainInfo  OPTIONAL,
      crossReferences     [1]    SEQUENCE OF CrossReference OPTIONAL,
                          [2]    SecurityParameters DEFAULT {}}
```

12.4.2 各种自变量的含义在 § 12.4.2.1 至 § 12.4.2.3 中定义。

12.4.2.1 自变量 info，用来向所有参与处理某一个共同请求的 DSA 传送特定 DMD 信息。此自变量属于 **DomainInfo** 类型，它是无限制类型。

12.4.2.2 除非相应请求中的自变量 **returnCrossRefs** 取 **TRUE** 值，否则在 **ChainingResults** 中不存在 **crossReferences** 自变量。此自变量包含 **CrossReference** 项序列，每项包含一个 **contexPrefix** 和一个 **accessPoint** 描述符(见 § 12.8)。

```
CrossReference        ::=    SET{
      contextPrefix   [0]    DistinguishedName,
      accessPoint     [1]    AccessPoint}
```

当 **CrossReference** 其中的一个上下文前缀与抽象操作的 **targetObject** 自变量的一部分匹配时，DSA 就可以把这个 **CrossReference** 加到序列中去。DSA 的行政权威的政策可能不要求回送上述知识，从而不需要

在序列中增加项目。

12.4.2.3 自变量 **SecurityParameters** 在建议 X.511 中定义。它的缺席相当于保密参数的集合是空的。

12.5 操作进展

12.5.1 **OperationProgress** 的值说明在有多个 DSA 参与的抽象操作中操作的进展状况。

```
OperationProgress ::= SET {
    nameResolutionPhase [0]
        ENUMERATED {
            notStarted    (1),
            proceeding    (2),
            completed     (3)},
    nextRDNToBeResolved [1]
        INTEGER OPTIONAL}
```

12.5.2 各自变量的含意在 §12.5.2.1 和 §12.5.2.2 中定义。

12.5.2.1 自变量 **nameResolutionPhase** 指出在处理某一操作中 **targetObject** 名的过程中达到的阶段。倘若 **nameResolutionPhase** 指的是 **notStarted**，则这表示这个时候未能达到某一个包含名称初始 RDN 的命名上下文的 DSA。倘若名称鉴别，指的是 **proceeding**，则表示已识别名称的初始部分，但未能达到包含目标客体的 DSA。**nextRDNToBeResolved** 表示名称已识别到什么程度（§12.5.2.2）。倘若名称鉴别指的是 **completed**，则表示已到达包含目标客体的 DSA，而操作本身仍在进行中。

12.5.2.2 自变量 **nextRDNToBeResolved** 向 DSA 指出，下一个要鉴别的是 **targetObject** 名中的哪一个 RDN。它在范围 1 内采用整数形式来表示 RDN 在名称中的总数。这个自变量仅在 **nameResolutionPhase** 取 **proceeding** 值的情况下才出现。

12.6 跟踪信息

12.6.1 **TraceInformation** 值传递参与操作的 DSA 的记录。它用来检测或避免循环，它是由于知识不一致或 DIT 中出现别名循环造成的。

```
TraceInformation    ::=    SEQUENCE OF TraceItem
TraceItem           ::=    SET {
    dsa                    [0] Name,
    targetObject           [1] Name OPTIONAL,
    operationProgress      [2] OperationProgress }
```

12.6.2 当 DSA 要把操作传播到另一个 DSA 时，该 DSA 就在跟踪信息里增添新项目。每一个 **TraceItem** 包含下列内容：

a) 增添新项目的 dsa 的名称 **Name**；

b) **targetObjectName**：即 DSA 由请求中收到的目标客体名。此参数在下列情况下可以省略：被连锁转移的请求是来自 DUA（在此情况下，其隐含值在 **XOperation** 中是 **object** 或 **baseObject**）；或者当它的值与输出请求中的（实际或隐含的）**ChainingArgument** 所包含的 **targetObject** 相同时；

c) **operationProgress**：即 DSA 由输入请求中收到的操作进展参数。

12.7 参考类型

12.7.1 **ReferenceType** 的值分别指出在 §10 中定义的各种参考。

```
ReferenceType ::=

    ENUMERATED {
        superior                (1),
        subordinate             (2),
        cross                   (3),
        nonSpecificSubordinate  (4)}
```

12.8 访问点

12.8.1 **AccessPoint** 的值表示访问号码簿或特定 DSA 的某一个接入点。后者用于与该 DSA 进行的 OSI 通信联系。

```
AccessPoint     ::=     SET {
        ae-title        [0]  Name,
        address         [1]  PresentationAddress }
```

12.9 连续参考

12.9.1 **ContinuationReference** 表示在一个或多个 DSA 中如何完成全部或部分抽象操作。倘若有关的 DSA 不能够或不愿意把请求传递给别的 DSA，则通常的作法是把它做为一个转移参考回送。

```
ContinuationReference   ::=   SET {
    targetObject        [0]   Name,
    aliasedRDNs         [1]   INTEGER OPTIONAL,
    operationProgress   [2]   OperationProgress,
    rdnsResolved        [3]   INTEGER OPTIONAL,
    referenceType       [4]   ReferenceType OPTIONAL,
```

12.9.2 各参数的含义在 §12.9.2.1 至 §12.9.2.6 中定义。

12.9.2.1 **targetObjectName**：建议用来继续操作。这参数可能与在输入请求中收到的 **targetObject Name** 不一样，例如若有一个别名已换名，或在搜索中找到了基础客体。

12.9.2.2 **aliasedRDN** 指出，在目标客体名中，通过更换别名产生了多少个 RDN(若存在的话)。仅当有一个别名已换名时此自变量才出现。

12.9.2.3 **operationProgress** 指出抽象操作已进展到什么程度，并指导被指名的 DSA 继续完成抽象操作(假定收到 **ContinuationReference** 的 DSA 或 DUA 能继续完成操作的话)。

12.9.2.4 **rdnsResolved** 的值指出，仅使用内部参考实际能鉴别的 RDN 个数。仅当名称中的一些 RDN 未成为完整的名称鉴别的主体，但根据交叉参考认为是正确时，本自变量才需要出现。

12.9.2.5 **referenceType** 自变量只出现于 DSA 抽象服务之中，它指出在生成此继续操作时使用的知识类型。

12.9.2.6 **accessPoints** 指出拟完成继续操作目标必须跟踪的访问点。在使用非特定下级参考的场合，列出的 **AccessPoint** 可能不止一个，而且每个访问点必须加以跟踪，例如采用多点传播方式。

13 抽象耦合与抽象释放

DSABind 和 **DSAUnbind**：DSA 在开始和结束访问其他 DSA 的过程中，分别使用 **DSABind** 和 **DSAUnbind**。

13.1 *DSA* 耦合

13.1.1 **DSABind** 抽象耦合操作：DSA 使用此自变量将 **chainedRead**，**chainedSearch** 和 **ChainedModify** 端口耦合到别的 DSA 的相应端口。

```
DSABind     ::=     ABSTRACT-BIND
    TO              {chainedRead,
                    chainedSearch,
                    chainedModify}
    DirectoryBind
```

13.1.2 **DSABind** 与 **DirectoryBind**(见建议 X.511)中的相应自变量是相似的,但有以下区别。

13.1.2.1 **DirectoryBindArgument** 的 **Credentials**(凭证)允许发起者 DSA 将它的 AE-Title 的识别信息送到响应者 DSA。AE-Title 必须采用号码簿可识别名的形式表示。

13.1.2.2 **DirectoryBindResult** 的 **Credentials** 允许响应者 DSA 将它的 AE-Title 的识别信息送到发起者 DSA。AE—Title 必须采用号码簿可识别名的形式表示。

13.2 *DSA 释放*

13.2.1 **DSAUnbind** 操作用于释放一对 DSA 的连锁读、连锁搜索和连锁修改端口。

```
DSAUnbind    ::=    ABSTRACT-UNBIND
    FROM            {chainedRead,
                     chainedSearch,
                     chainedModify}
```

13.2.2 这里没有自变量、结果和差错。

14 连锁抽象操作

14.1 对应于号码簿抽象服务的每一个端口是 DSA 的一个端口;相互合作的 DSA 通过这个端口提供抽象服务。在每一个端口上的抽象操作也是有一个对应关系。端口名和抽象操作名的选择也应该是为了反映这种对应关系,DSA 抽象服务的端口或抽象操作是由号码簿抽象服务中的相应部分冠以"前缀"构成的。上述端口和抽象操作的最终形式如下:

```
ChainedReadPort:      ChainedRead,
                      ChainedCompare,
                      ChainedAbandon

ChainedSearchPort:    ChainedList,
                      ChainedSearch

ChainedModifyPort:    ChainedAddEntry,
                      ChainedRemoveEntry,
                      ChainedModifyEntry,
                      ChainedModifyRDN
```

14.2 连锁操作的自变量、结果和差错分别由号码簿抽象服务(如 § 14.3 中描述的)之中的对应抽象操作的自变量、结果和差错构成,只有一个例外。这个例外是 **ChainedAbandon** 抽象操作,它与号码簿抽象服务的对应部分(如 § 14.4 中所述)在语法上是等效的。

14.3 **ChainedX** 抽象操作用于在 DSA 之间传播一个请求;这个请求(通常)是来自当一个 DUA 在一个 DSA 上起动 X 抽象操作时,该 DSA 选择采用连锁方式进行传递。抽象操作的自变量可由起动者选择是否要署名;倘若要求署名,则参与操作的 DSA 可以在结果上署名。

14.3.1 连锁抽象操作 **Chained X** 从其对应部分 X 的推导如下:

给出:

```
X    ::=
     ABSTRACT-OPERATION
          ARGUMENT   XArgument
          RESULT     XResult
          ERRORS     {..., Referral,...}
```

连锁抽象操作的推导过程如下：

```
ChainedX  ::=
    ABSTRACT-OPERATION
        ARGUMENT OPTIONALLY-SIGNED SET{
                ChainingArgument,
                [0]  XArgument}
    RESULT  OPTIONALLY-SIGNED SET{
                ChainingResult,
                [0] XResult}
    ERRORS {...,DsaReferral,...}
```

注 — 附件 A 中的 DSA 抽象服务的定义规范充分利用上述推导方法来描述连锁抽象操作。

14.3.2 导出的抽象操作的自变量的含义在 § 14.3.2.1 和 14.3.2.2 中描述。

14.3.2.1 **ChainingArgument** 包含 DSA 完成操作所需要的信息(除发起者 DUA 提供的自变量之外)。此信息类型在 § 12.3 中定义。

14.3.2.2 **XArgument** 包含发起者 DUA 提供的自变量，后者在建议 X.511 中的相应条款中定义。

14.3.3 倘若申请获得成功，则应回送结果。结果参数的含义在 § 14.3.3.1 和 § 14.3.3.2 中描述。

14.3.3.1 **ChainingResult** 包含的信息(除为发起者 DUA 提供的信息之外)是链中的诸 DSA 所需要的。此种信息类型在 § 12.4 中定义。

14.3.3.2 **XResult** 包含由完成本抽象操作者回送的结果，并准备在结果中反馈给发起者 DUA。此信息的定义在建议 X.511 的相应条款中描述。

14.3.4 倘若请求失败，则应回送所列差错之一。差错通知中的差错信息在建议 X.511 中对应抽象操作定义中描述，唯此地回送的转移参考是 **DSAReferral** 而不是 **Referral**。各种差错在 § 15 中定义或参照。

14.4 **ChainedAbandon** 抽象操作的用途：当 DSA 对原先申请起动的连锁操作不再感兴趣时，它就可以使用此信息指向别的 DSA。其理由可能很多，下面举几个例子：

a) 导致该 DSA 提出连锁操作的操作本身已经被放弃，或者因断绝联系而隐含地被中断；

b) DSA 已通过别的渠道获得所需要的信息，例如从参与多点传播的快速响应 DSA 中获得。

DSA 没有义务去发出 **ChainedAbandon**，或者放弃操作(如果被要求这样做的话)。

倘若 **ChainedAbandon** 在停止某一操作的尝试中获得成功，则应回送一个结果，然后由主操作回送一个 **Abandoned** 抽象差错。倘若失败，则它本身应负责回送一个 **AbandonFailed** 差错。

15 连锁抽象差错

15.1 绪论

15.1.1 就绝大部分情况而言，在号码簿抽象服务中回送的抽象差错，同样可以在 DSA 抽象服务中回送。所不同的是回送 **DSAReferral**“差错”(见 § 15.2)而不是 **Referral**，而且下列服务问题具有同样的抽象语法，但不同的语义学。

a） **invalidReference**

b） **loopDetected**

15.1.2 可能发生的抽象差错的先后次序，与在建议 X.511 中描述的号码簿抽象服务中的情况相似。

15.2 *DSA 转移参考*

15.2.1 当 DSA 由于某种原因不愿意采用连锁方式或多点传播方式将抽象操作传递给一个或多个 DSA 来继续完成某一个抽象操作时，它就生成 **DSAReferral** 抽象差错。DSA 回送一个转移参考的条件在 § 8.4 中描述。

```
DSAReferral    ::=
    ABSTRACT-ERROR
        PARAMETER SET{
            [0] ContinuationReference,
            contextPrefix [1] DistinguishedName OPTIONAL }
```

15.2.2 各种参数的含义在 § 15.2.2.1 和 § 15.2.2.2 中说明。

15.2.2.1 **ContinuationReference** 包含发起者继续向别的 DSA 传播进一步请求时所需要的信息。本信息类型在 § 12.9 中描述。

15.2.2.2 倘若本抽象操作使用的 **ChainingArguments** 中的 **returnCrossRefs** 取 **TRUE** 值，而且使用的转移参考是根据一个下级或交叉参考定义的，则 **contextPrefix** 参数可以任选。任何一个 DSA 的行政权威都能够决定哪些知识参考（若存在）可以使用这种方式回送（其他知识参考，对该 DSA 而言可能是保密的）。

第 5 部分 — *分布式操作规程*

16 绪论

16.1 *范围与界限*

本节描述由 DSA 完成的号码簿分布式操作的规程。每一个 DSA 单独执行下列规程：号码簿向用户提供的全套服务是由所有 DSA 成员的集体行动产生的。

为单个 DSA 提供的规程是根据本建议 § 7 至 § 10 中的模型描述的。

必须指出，这里所述的模型和规程只是为了说明目的而不是为了约束或指导实际 DSA 的实现。

本节共有三个子节：即绪论、描述号码簿性能的概念模型以及以 DSA 为核心和以操作为核心两种 DSA 操作模型的介绍。

16.2 *概念模型*

由于号码簿分布式操作非常复杂，故需要采用文字说明与图解技术来建立一个概念模型。但是，不论是文字说明还是示意图都不能理解为分布式号码簿操作的形式描述。

16.3 *DSA 的个别操作与集体操作*

模型是从两个分开的角度看 DSA 操作的；两个合起来就能提供一幅完整的号码簿操作图。

a) 以 DSA 为核心的模型:在这个模型中,支持号码簿的一套规程是从一个 DSA 的观点描述的。这个模型能够为每一个规程提供确切的描述,同时能够全面描述这些规程之间相互关系和总体控制结构。第 18 节从以 DSA 为核心的角度描述 DSA 规程。

b) 以操作为核心的模型:以 DSA 为核心的模型能够提供完整的操作说明的细节,但是要理解个别操作的结构是困难的,因为要由多个 DSA 进行处理。因此,§17 采用主要以操作为核心的模型来介绍适用于每一个 DSA 的处理阶段。

为了支持号码簿的分布式操作,每一个 DSA 必须参与某些行动来实现每一个操作的意图,和参与完成某些附加行动来向多个 DSA 传播实现该操作的请求。§17 探讨这两种行动的区别。§18 详述这两种行动。

17 分布式号码簿的性能

17.1 通过协作完成的操作

每一个 DSA 都有一套规程,使它能够全面完成全部号码簿操作。在 DSA 拥有整个 DIB 的情况下,全部操作实际上都在该 DSA 内部完成。在 DIB 分布于多个 DSA 的情况下,全部操作是分段完成的,每一部分操作分别在每一个相互合作的 DSA 内部完成。

在分布式环境里,DSA 把每一个操作视为暂时的事件;操作是由一个 DUA 或其他 DSA 起动的;DSA 在客体上进行操作,然后把它传给别的 DSA 继续处理。

另一种观点认为,全部操作是由多个相互合作的 DSA 共同完成的。这个观点揭示了适合于所有操作的通用处理阶段。

17.2 操作处理阶段

每一个号码簿操作都可设想分为三个不同的阶段:

a) 名称鉴别阶段:在这个阶段,利用客体名(该客体的登录项是特定操作的对象)去寻找储存登录项的 DSA;

b) 评价阶段:在这个阶段,由特定号码簿请求指定的操作(例如,读)已经正式进行;

c) 结果合并阶段;在这个阶段,特定操作的结果被送回请求者 DUA。倘若选择了连锁操作方式,结果合并阶段可能包括多个 DSA,每一个 DSA 在上述一个或两个阶段中把原请求或子请求(在 §17.3.1 的请求分解中所定义的)连锁转移到其他 DSA。

在 **Read**,**Compare**,**List**,**Search** 和 **ModifyEntry** 操作情况下,名称鉴别是在操作自变量里提供的客体名上进行的。在 **AddEntry**,**RemoveEntry** 和 **ModifyRDN** 操作情况下,名称鉴别是在直接上级客体的名称上进行的,即通过在操作自变量里提供的名称中删除最后的 RDN 获得的。

对于特定登录项的操作,从一开始就可以指定给号码簿里的任何 DSA。该 DSA 使用其知识(也许和其他 DSA 一起)处理三个阶段的操作。

17.2.1 名称鉴别阶段

名称鉴别是把声称名中的每一个 RDN 与 DIT 中的一个弧线(或极点)顺序对照,逻辑上是从 DIT 中的根开始往下进行。但是,由于 DIT 是在数量不限的 DSA 之中分布,因此每一个 DSA 也许只能够完成名称鉴别进程的一小部分。一给定 DSA,通过它的本地知识完成名称鉴别进程的一部分。上述进程在 §18.6 和附图(图 11/X.518 至图 13/X.518)中描述。当 DSA 到达它的命名上下文的边界时,它就能够从其中的知识信息中知道另一个 DSA 是否能够继续进行名称鉴别,或者知道该名称是否错误。

17.2.2 评价阶段

当名称鉴别阶段结束之后，实际要求的操作(如读或搜索)就开始进行。

只涉及一个登录项的操作—**Read**、**Compare**、**AddEntry**、**ModifyRDN** 和 **ModifyEntry**—完全可以在包含该登录项的 DSA 内部完成。**AddEntry**、**RemoveEntry** 和 **ModifyRDN** 有可能影响不止一个 DSA 中的知识。参见 § 18.7.1。

涉及多个登录项的操作—**List** 和 **Search**—需要寻找目标的下级，该下级可能留在同一个 DSA 之中。倘若它们不是全部留在同一个 DSA 之中，则需要把操作传给在下级参考中指出的 DSA，以完成上述评价进程。

17.2.3 结果合并阶段

一旦获得评价阶段的结果就可以进入结果合并阶段。

在仅涉及到一个登录项的操作中，操作结果可以直接回送给请求者 DUA。在涉及多个 DSA 的多个登录项的操作中，需要把结果合并在一起。

结果合并之后，允许回送给请求者的响应包括下列几项：

a) 全部操作结果；

b) 不完整的结果：由于 DIT 的某些部分仍然处于未被探索阶段(仅适用于 **List** 和 **Search** 操作)所致。这样的部分结果可以包含继续参考来指出 DIT 中未被探索的部分；

c) 一个差错(其特例是一个转移参考)；

d) 一个连锁结果(**ChainingResult**)，若请求者为 DSA。

17.3 分布式操作的管理

要求 DSA 完成的每一个抽象操作使用的自变量，应当包含适当的情报，指出每一个操作在通过号码簿的各个 DSA 的过程中的进展情况。这样，每一个 DSA 就能完成交给它的那部分任务，并且在它把操作交给别的 DSA 继续完成之前先把完成的那一部分操作记录下来。

DSA 的附加规程允许 DSA 在物理上把操作分配给其他的 DSA，并支持由上述分配引起的其他需求。

17.3.1 请求的分解

请求分解是 DSA 在同其他的 DSA(一个或多个)进行交互式操作之前需要完成的内部进程。一个请求可分解为若干子请求，使得每一个子请求完成总任务的一部分。举一个例子，在找到基础客体之后，接着就可以使用请求分解进行搜索操作或其他操作。在完成请求分解之后，就可以把每一个子请求连锁转移或多点传播给其他的 DSA，以继续完成任务。

17.3.2 响应请求的 DSA

收到某一个请求的 DSA 可借助于操作进展参数检查请求的进展情况。根据上述检查结果，DSA 就可以判断操作是仍然停留在名称鉴别阶段，还是已经到达评价阶段，并确定 DSA 应当设法满足的操作部分。倘若 DSA 无法全部满足该请求，则它必须(通过连锁转移或多点传播)把该操作传给能协助满足该请求的一个或多个 DSA，或向另一个 DSA 回送转移参考，或使用差错来中止该请求。

17.3.3 操作的结束

起动操作或将操作传播给一个或多个 DSA 的每一个 DSA 必须对操作情况进行跟踪，直到其他 DSA 都分别回送一个结果或差错信息，或超过操作的最大时限为止。上述要求对所有的操作、传播方式和处理阶段都有效。上述措施保证已经在号码簿里传播的分布式操作能够有条不紊地告一段落。

17.4 有关分布式操作的其他考虑

17.4.1 对请求的检验

当 DSA 接受号码簿操作时，首先必须对该操作进行检验，以保证操作顺利进行。因使用不恰当的别名或错误的知识在 DIT 内部造成的循环有可能使送到 DSA 的操作无法处理。

在简单的实例中，可使用 § 18 所述的名称鉴别规程对这些差错进行充分处理。但是，在导致产生操作循环的场合(如 § 17.4.3 所述)，光使用名称鉴别规程是不够的。

对请求进行检验，保证在使用因循环造成的错误数据进行操作之前能够及早发现循环。循环检测是采用 § 18.5.1 中所述的循环检测规程进行的。

在实施安全保密措施的地方，可使用请求检验规程对提出请求的 DSA 或 DUA，以及请求的合法性进行检验。

17.4.2 状态与跟踪信息

在号码簿内的操作进展情况和循环条件的存在，都可以通过一个操作“状态”来确定。该状态的定义如下：

— 当前正在处理操作的 DSA 的名称；

— 包含在操作自变量内的 **targetObject** 名；

— 包含在操作自变量内并在 § 12.5 中定义的 **operatingProgress**。

除了当前的操作状态之外，DSA 还需要知道以前的全部操作状态。这些状态都记录在 **traceInformation** 自变量之中，并在操作过程中传递。

自变量 **traceInformation** 是 § 17.4.3 中所述的防止/检测循环策略的基础。

17.4.3 循环现象

在特定号码簿操作的上下文中，若操作始终返回原先的状态(如前面定义的)，则表示循环现象已经发生。对循环可用 **traceInformation** 自变量进行控制。目前定义了两种控制循环的策略。在循环检测中，DSA 首先判断本操作是否已出现循环现象。若是，则回送一个差错。在循环预防措施中，DSA 应判断某操作(若传给别的 DSA)是否会产生循环。

17.4.4 服务控制

在分布式环境中，有些服务控制需要特殊考虑，以便能够按照所要求的方式完成操作。

a) **ChainingProhibited**(禁止使用连锁方式)：在确定使用哪一种传播方式之前，DSA 首先检查本服务控制。若本服务控制被置位，则 DSA 应始终使用转移参考方式；否则，DSA 可选择采用连锁方式或

转移参考方式，视其功能而定。

b) **timeLimit**(时限)：DSA 必须考虑本服务控制以确保在该 DSA 中的操作不超过时限。当 DUA 向某一个 DSA 提出完成某种操作的请求时，该 DSA 首先必须注意到，DUA 提供的 **timeLimit** 是完成该项操作的时间限制(以秒计)。倘若需要采用连锁方式，则应把 **timeLimit** 放入拟传给另一个 DSA 或多个 DSA 的连锁自变量。在此情况下，每个连锁请求都使用同一个时限值，即 UTC 通用协调时间，并且务必在此时间内完成操作来满足原先给出的限制条件。一旦收到包含时限 **timeLimit** 的连锁自变量，DSA 就应当遵守这个限制条件。

c) **sizeLimit**(限额)：DSA 必须考虑本服务控制，以确保结果表不超过规定的范围。在采用连锁方式或多点传播方式传递请求的过程中，原请求中的自变量包含的时限应保持不变。倘若需要把请求加以分解，则应在拟传给下一个 DSA 的自变量中放入同样的时限：即每个子请求都使用全时限。当结果被回送时，请求者 DSA 把多个结果加以分解，并限制其总数，以保证只回送所申请的数量。倘若超过了限额，则应在应答中指出。

d) **Priority**(优先级)：在所有的传播方式中，每个 DSA 必须确保对操作的顺序处理以支持本服务控制(若存在的话)。

e) **localScope**(本地范围)：操作仅限于本地定义的范围，而不能以任何方式进行传播。

f) **scopeOfReferral**(转移参考的范围)：倘若 DSA 向一个 **List** 或 **Search** 操作回送一个转移参考或部分结果，则嵌入的 **ContinuationReferences** 必须是在所要求的范围内。

所有其他的服务控制都必须受到尊重，但是它们在分布式环境中的应用并不要求任何特殊考虑。

17.4.5 扩展

17.4.5.1 倘若 DSA 在名称鉴别处理阶段碰到一个扩展的抽象操作，而且确定该抽象操作必须以连锁方式传给另一个或多个 DSA，则该 DSA 必须把遇到的任何扩展原封不动地放入该连锁抽象操作。

注 — 倘若行政权威不希望把扩展部分向外传播，则它可以决定回送一个 **ServiceError**，并指出 **unwillingToPerform** 的问题。

17.4.5.2 倘若 DSA 在执行阶段遇到一个扩展的抽象操作，则有两种可能性。第一种可能性是扩展的操作不很关键，因此 DSA 可以不理睬。第二种可能性是该扩展部分很关键，这时 DSA 应回送 **ServiceError**，并指出 **unavailableCriticalExtension** 的问题。

在多客体操作中，关键性扩展会导致这种类型的结果差错和服务差错。合并上述结果和差错的 DSA，应放弃这些服务差错，并使用建议 X.511 §10.1.1 所述的 **PartialOutcomeQualifier** 中的 **unavailableCriticalExtension**。

17.4.6 更换别名

更换别名是建立新目标客体名的过程，即用别名登录项中的别名客体名属性值来代替原目标客体名的别名登录项中的可识别名。在操作中的客体名不受更换别名的影响。

17.5 分布式操作的验证

号码簿用户及其行政权威可随意要求验证号码簿操作。对任何特定号码簿操作而言，验证方法的种类均取决于实施的安全保密政策。

有两套验证规程可以满足一系列验证要求。第一套规程由耦合规程提供:这套规程适用于两个号码簿应用实体之间的验证,其目的是建立联系。耦合规程包括一系列验证法,从简单地互相验证身份证到强验证法。

除了耦合为同级实体的联系提供验证规程之外,另外还在号码簿里定义附加规程,允许对个别操作进行验证。现已定义了两套号码簿验证规程。第一套规程是为了便于对发起者进行验证,即为DSA对原服务请求进行验证提供的规程。第二套规程是便于对结果进行验证,即为发起者对回送的任何结果进行验证提供的规程。

为发起者定义的两套验证规程:其一,叫做身份证验证规程,采用身份证的简单交换;其二,叫做签署式验证规程,采用数字签署技术。前者的形式很简单,是采用公开传输的可识别名实现的。

在结果的验证方面,只定义了一套结果验证规程。该规程采用数字签署技术。由于结果的整理通常很复杂,故没有定义更简单的身份证验证规程。

这些规程不支持差错响应的验证。

上述服务是对耦合服务的补充;在对号码簿操作进行验证之前,假设耦合规程已经生效。

DSA提供发起者和结果验证使用的规程在§18.9中描述。

18 DSA的特性

18.1 绪论

当请求者(例如DUA或DSA)每次起动操作时,负责操作的DSA必须遵照明确定义的规程进行操作,以保证对方回送适当的响应。本节利用实现某些规程的进程建立一个DSA模型,来说明它的特性。必须指出,对DSA只要求符合相关规程所表现的外部特性而不是规程本身。

18.2 *DSA*特性概述

分布式号码簿的总特性是所有参与操作的协作DSA的特性的总合。每一个DSA可看做由内部规程支持的一个进程。

图6/X.518表示DSA特性的内部示意图。

操作调度,在DSA中是最主要的控制规程。它把每一个操作引到§17.2所述的三个处理阶段。

支持操作调度的规程是:名称鉴别、寻找名称上下文、本地名称鉴别、评价、单客体评价、多客体评价和结果合并规程。这些规程之间的关系示于图6/X.518。

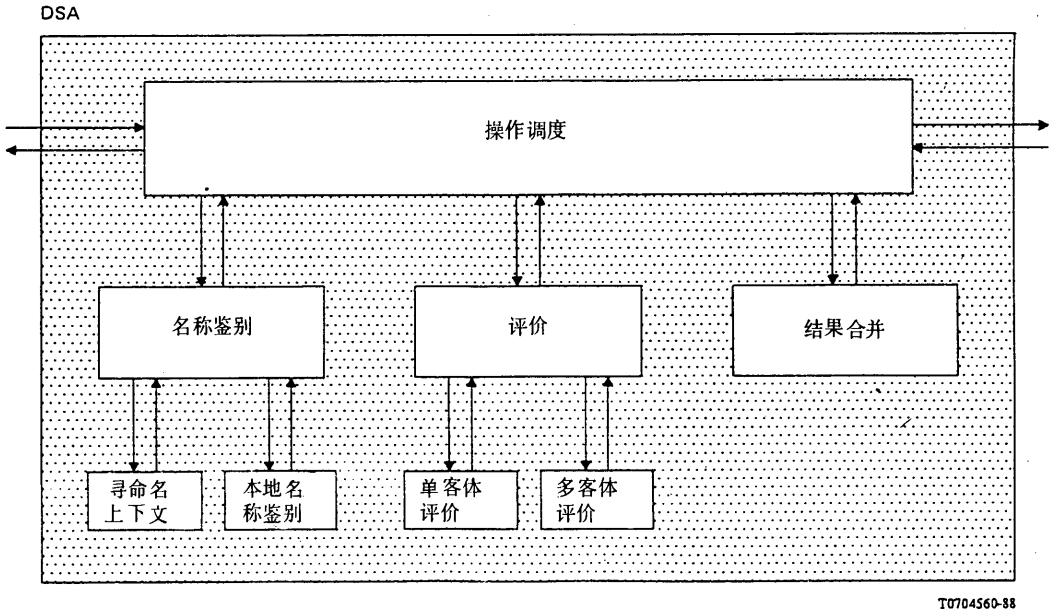


图 6/X.518

DSA 特性的内部示意

18.2.1 操作调度

一开始接受操作时，操作调度首先对它进行验证，检查是否产生循环或有验证差错。若检验结果没有发现任何差错，则调度可着手调用名称鉴别规程，后者将回送一个“找到”指示、一个参考或一个差错指示。参考是由一个转移参考，或一个连锁操作，或多点传播操作负责处理的，“找到”指示则是通过调用实际完成所希望的那个操作的评价规程处理的。一旦回送内部或外部结果，它们便由结果合并进行整理，并在无差错的情况下把它们回送给主叫 DUA 或 DSA。

18.2.2 名称鉴别规程

名称鉴别规程调用寻找命名上下文。倘若被回送的上下文属于本地，则调用本地命名鉴别规程，否则名称鉴别规程应回送一个参考或一个差错信息，然后结束。倘若本地名称鉴别规程遇到一个别名，则应更换别名(若允许)，然后由名称鉴别规程从头开始重复做分析。否则名称鉴别规程将回送一个“找到”指示、一个差错指示或一个转移参考，并把它送回操作调度。

18.2.3 寻找命名上下文

寻找命名上下文设法将声称名与上下文前缀进行对照。倘若都不匹配，则寻找命名上下文设法对交叉参考或上级参考进行识别。倘若找到一个匹配的上下文前缀，则寻找命名上下文将回送指向 DIT 下方的交叉参考，或者回送一个指示，表示在本地找到了 一个合适的命名上下文，并把 **NameResolutionPhase** 置为“进展(proceeding)”。

18.2.4 本地名称鉴别

本地名称鉴别规程设法在声称名内部找到合适的 RDN，直至能够回送“找到”指示为止。倘若无法在内部匹配所有的 RDN，则本规程首先着手识别特定下级参考，然后识别非特定下级参考，并将上述参考回送给名称鉴别规程。若遇到一个别名，而且服务控制允许它换名，则应回送一个别名换名指示。若不允许换名，则

仅当遇到别名时所有的 RDN 都已经换名，才回送一个“找到”指示，否则应回送 **nameError**。

18.2.5 评价规程

根据请求，评价规程实际执行对目标客体的号码簿操作，根据操作的类型，分别起动单客体评价规程或多客体评价规程。

18.2.6 单客体评价规程

单客体评价规程是针对 **Read**，**Compare**，**AddEntry**，**RemoveEntry**，**ModifyEntry** 和 **ModifyRDN** 起动的。在这个规程中才真正需要对属性进行检索、检验或修改。

18.2.7 多客体评价规程

多客体评价规程是在需要进行 **Search** 和 **List** 操作时起动的，目的是为了检验过滤器、检索结果和必要时调度子请求。

18.2.8 结果合并规程

结果合并规程是用来把从其他 DSA 收到的结果或差错同在本地检索到的结果加以核对。

18.3 特定操作

操作可分为三种类型(每一种操作及其连锁操作的对应部分都属于同一类型)：

a) 单客体操作：**Read**、**Compare**、**AddEntry**、**ModifyEntry**、**ModifyRDN**、**RemoveEntry**。

b) 多客体操作：**List**、**Search**。

c) 放弃操作：**Abandon**。

上述类型的处理方法分别在 § 18.3.1 至 § 18.3.3 中描述。由于 DSA 在进行服务端口操作和连锁服务端口操作时所表现的特性很相似，故只需提供一种适合于两种操作的描述，并指出其中的一些差别。

18.3.1 单客体操作

单客体操作只涉及一个登录项，从而完全可以在包含操作对象登录项的 DSA 内完成。这些操作可用下列事件统一描述：

1) 激活操作调度。

2) 执行名称鉴别规程，以便寻找在操作自变量中指名的客体。

3) 执行单客体评价规程。

4) 在操作过程中，必须对诸如时限等服务控制进行检查，以满足用户提出的限制条件。

5) 向发送请求的 DSA 或 DUA 回送结果。

18.3.2 多客体操作

多客体操作涉及多个登录项，这些登录项可能都在一个 DSA 中，但也可以例外。因此，这样的操作照例是由多个协作的 DSA 共同完成，例如寻找所有相关的登录项和对它们进行操作。上述操作的共同特性描述如下：

1） 激活操作调度。
2） 执行名称鉴别规程，以便寻找在操作自变量中指名的客体。
3） 一旦找到操作目标客体，执行多客体评价规程。
4） 若请求分解已在多客体评价规程之一完成，而且已把子请求通过连锁或多点传播方式传给别的DSA，则操作调度保存当前的本地结果，等待连锁响应并激活结果合并规程。
5） 服务控制（如时限、限额）必须在操作过程中检查，保证不超过在自变量中提出的限制条件。
6） 向发送请求的 DSA 或 DUA 回送结果或差错。

18.3.3 放弃操作

一旦收到一个放弃操作的请求，DSA 应确定是否能够放弃指定的操作，若能则回送一个结果（被放弃的操作则回送一个 **Abandoned** 差错指示）。若不能放弃指定的操作，则 DSA 回送一个 **AbandonedFailed** 差错指示。

有关 **Abandon** 操作的规程描述如下：

1） 寻找在 **Abandon** 操作自变量中指出的起动标识符所对应的操作。
2） 可选项：利用适当的起动标识符组织一个（或多个）请求，申请放弃待执行的对其他 DSA 的连锁操作或多点传播操作。
3） 可选项：在内部完成放弃操作，如建议 X.511 所阐述的。
4） 向发送请求的 DSA 或 DSA 回送结果或差错信息。

18.4 操作调度

8.4.1 绪论

操作调度使用本建议 § 18.6 所述的名称鉴别规程，以及在分布式号码簿环境中寻找目标登录项所需的全部交互式操作（即 DSA 到 DSA 或 DUA 到 DSA）。图 7/X.518 表示操作调度详细流程图。其算法在下面概述。

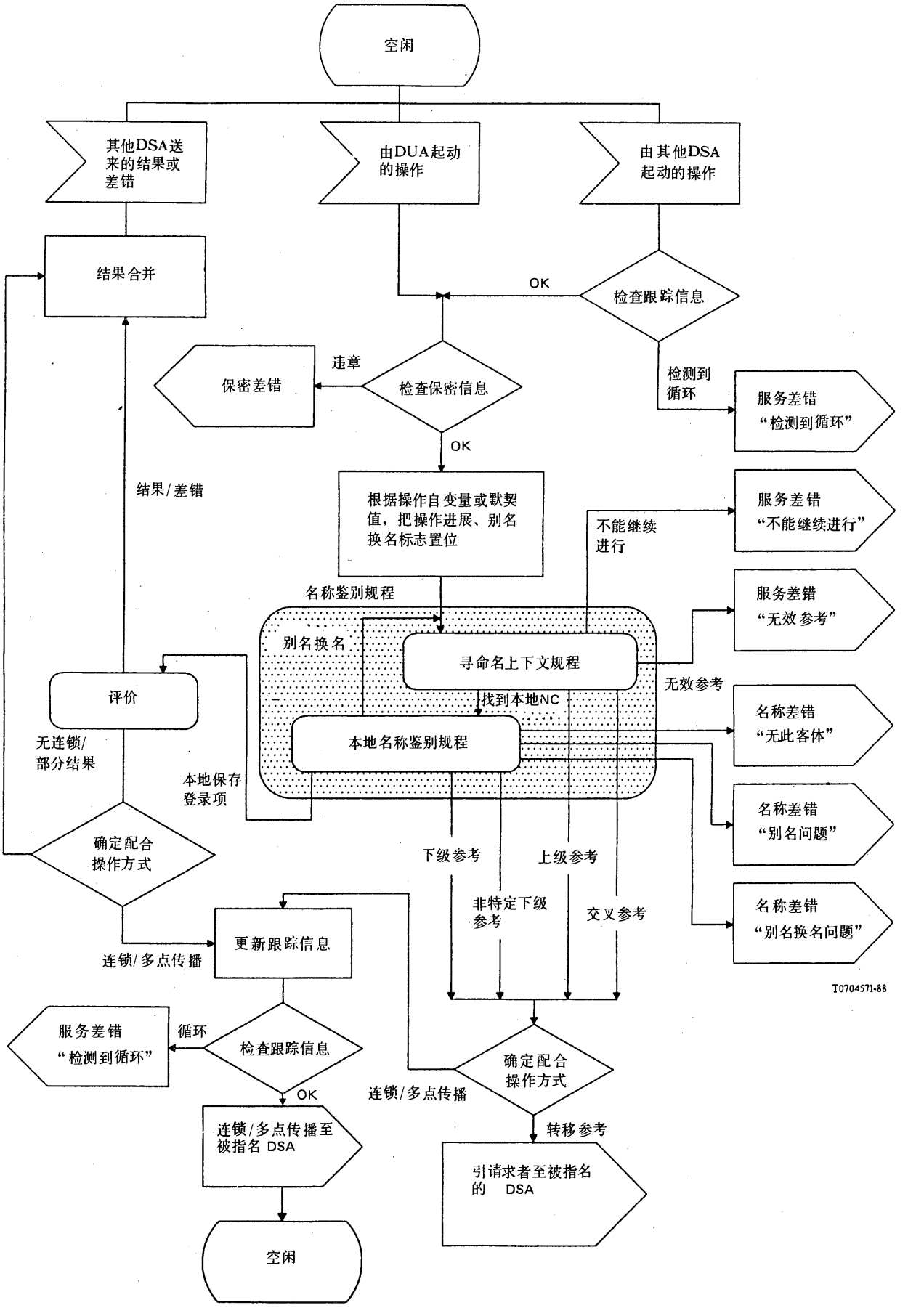

空闲
其他DSA送来的结果或差错
由DUA起动的操作
由其他DSA起动的操作
结果合并
OK
检查跟踪信息
保密差错
违章
检查保密信息
检测到循环
服务差错
“检测到循环”
OK
结果/差错
根据操作自变量或默契值，把操作进展、别名换名标志置位
不能继续进行
服务差错
“不能继续进行”
名称鉴别规程
别名换名
寻命名上下文规程
服务差错
“无效参考”
无效参考
评价
找到本地NC
名称差错
“无此客体”
本地名称鉴别规程
无连锁/部分结果
本地保存登录项
名称差错
“别名问题”
确定配合操作方式
下级参考
上级参考
非特定下级参考
交叉参考
名称差错
“别名换名问题”
更新跟踪信息
连锁/多点传播
T0704571-88
服务差错
“检测到循环”
循环
检查跟踪信息
确定配合操作方式
OK
连锁/多点传播
连锁/多点传播至被指名 DSA
转移参考
引请求者至被指名的 DSA
空闲

图 7/X.518

操作调度

18.4.2 隐含动作

18.4.2.1 安全保密措施

必须指出，虽然在算法里未包括对签字的检验，但是，当署名的操作、结果或差错到达 DSA 时，这应该永远是首先要采取的动作。

注 — 这不包括嵌套署名在内。

倘若签字无效，或在必须存在的地方却不存在，则回送一个 **SecurityError** 信息。所有的操作处理都被中止，操作调度回到空闲状态。

若要求签署操作结果，则在发送该结果之前它也是一个隐含的最后步骤。

18.4.2.2 服务控制

虽然不明显表示 **ServiceControls**，但仍然应当引起重视。例如，对到达操作的 **timeLimit** 的检验和发送结果之前对 **sizeLimit** 的检验都认为是强制性的。上述服务控制在 § 17.4.4 中讨论。

18.4.2.3 跟踪信息

TraceInformation 在放入 **ChainingArgument** 之前，始终使用到达 **DSA** 时的状态进行更新。对此下文并未明显表示。

18.4.3 自变量

为特定操作指定的连锁自变量。

18.4.4 结果

特定操作的连锁结果。

18.4.5 差错

在本建议中定义的任何差错。

18.4.6 算法

1) 接受操作
   倘若操作是从另一个 DSA 开始的，则它将包含连锁自变量，包括：**operationProgress**，**aliasDereferenced**，**aliasedRDN**，**targetObject Name** 和 **traceInformation**，以及在原操作中包含的参数。
   倘若操作是从一个 DUA 开始的，则它将不包含 **aliasDereferenced** 指示：从而取 **FALSE** 值。自变量亦不包含任何 **TraceInformation**，因此不需要做循环检查。将 **targetObject Name** 置为操作中的目标客体名(见 § 17.2)。其他连锁自变量根据 DAP 操作中的参数设置。**Originator** 被置成用户名。

2) 倘若操作来源于 DSA，则必须检查跟踪信息以发现循环(激活循环检测程序)。倘若检测到循环，则回送 **ServiceError** 和 **LoopDetected** 问题并中止处理。

3) 对操作进行安全检查(不管该操作是来源于 DUA 或 DSA)。若发现违章现象，则回送一个 **SecurityError**。否则，根据操作自变量或默认值将 **operationProgress** 和 **aliasDereferenced** 置位。

4） 执行名称鉴别规程。
名称鉴别规程将回送一个“找到”指示、一个远端参考，或一个差错指示。

5） 可能出现的差错如下：
ServiceError(UnableToProceed)— 倘若 DSA 确定它收到的操作请求不属于它所保存的信息。
ServiceError(invalidReference)— 倘若 DSA 确定使用的知识参考无效。
NameError(noSuchObject)— 倘若发现在操作请求中指出的声称名是无效的。
NameError(aliasProblem)— 倘若更换别名之后发现该别名没有对应的客体。
NameError(aliasDereferencing Problem)— 倘若在不允许别名存在的情况下遇到一个别名。
一旦收到上述差错指示，操作调度应中止操作，并向发起分布式操作的 DUA 或 DSA 回送一个差错指示。

6） 倘若回送“找到”(Found)，则应激活评价规程。

7） 倘若回送远端参考(由名称鉴别或评价规程)，它可能为下列参考之一：交叉参考、下级参考、上级参考、或非特定下级参考。

倘若回送上述参考，则表示名称鉴别或评价规程无法在本 DSA 内完成，而必须由参考中指出的 DSA 配合完成。
然后，由操作调度检查转移参考或连锁方式。

8） 倘若选择转移参考方式或交互操作方式，则根据 **scopeOfReferral**，可向发送原请求的 DUA 或 DSA 回送信息作为转移参考，该信息包含在被回送的参考中；或者回送 **outOfScopeServiceError**。然后结束本操作的处理。

注 — 倘若 **returnCrossRefs** 取真值，而且有关的参考不是非特定下级参考或上级参考，另行政权威愿意提供知识，则转移参考中的上下文前缀可以置位。

9） 倘若选择连锁操作方式，则应将操作传给参考中指出的 DSA。在非特定下级参考的情况下，应将操作传给在非特定下级参考中指名的每一个 DSA。传递的方法有两种：多点传播方式和连锁传递方式。

10） 对每一个拟发送的操作应进行防循环检查。若发现防循环检查规程不适用，或未检测到循环现象，则应给连锁自变量分配一些值，包括跟踪信息的更新版本，然后发送该操作。

倘若(由于发生循环问题)不发送操作，则应回送一个服务差错指示(连同所发现的问题：检测到循环)，并中止本操作的处理。

注 — 倘若由于在此阶段采用防循环措施而放弃分解操作，关于是否要回送一部分结果或放弃整个操作以及回送差错指示等均属本地决策。倘若选择后一种，则应回送 **ServiceError** 服务差错指示(连同发现的问题：**loopDetected**)并中止处理。

11） 等待响应，然后执行“结果合并”规程。

18.5 循环操作

在特定号码簿操作的上下文内，若该操作随时返回原状态(如 § 17.4.2 所述)，则表明循环现象已经发生。这并不意味着一个特定 DSA 不能多次处理某一个操作，而表示 DSA 不愿在同一个状态下多次处理同一个操作。

处理循环现象可采用§12.6中定义的跟踪信息自变量。确定循环现象有两种策略:循环检测和防循环策略,分别在§18.5.1和§18.5.2中描述。

18.5.1 循环检测

循环检测要求,当DSA收到一个操作时,确定本操作的当前状态是否出现在操作自变量**traceInformation**记录下来的状态序列中。若出现,则操作是循环的,故应回送一个**ServiceError**(连同问题:**loopDetected**)。否则,DSA应遵照§18.4所述的规程继续进行处理。

18.5.2 防循环

防循环规程要求DSA,在它把某一个操作传给另一个DSA之前(做为连锁转移、多点传播、或分解请求规程的一部分),首先确定,该操作的相应状态(若已知)是否出现在原操作的跟踪信息自变量记录下来的状态序列中。该相应状态是接收方DSA准备加到**TraceInformation**的**TraceItem**的值。

在原输入操作到达一个服务端口(而不是一个连锁服务端口)的场合,是不会有跟踪信息的,因此防循环规程与之无关。

倘若操作的相应状态为已知,而且出现在**traceInformation**内部,则起动操作就会产生循环。在此情况下,给原操作回送的适当响应是**ServiceError**(连同问题:**loopDetected**)。

18.6 名称鉴别规程

本节详述名称鉴别规程,其输入和输出参数以及可能的差错情况。图7/X.518表示总规程的流程图。名称鉴别规程调用两个子规程:

1) 寻找命名上下文(图8/X.518)。

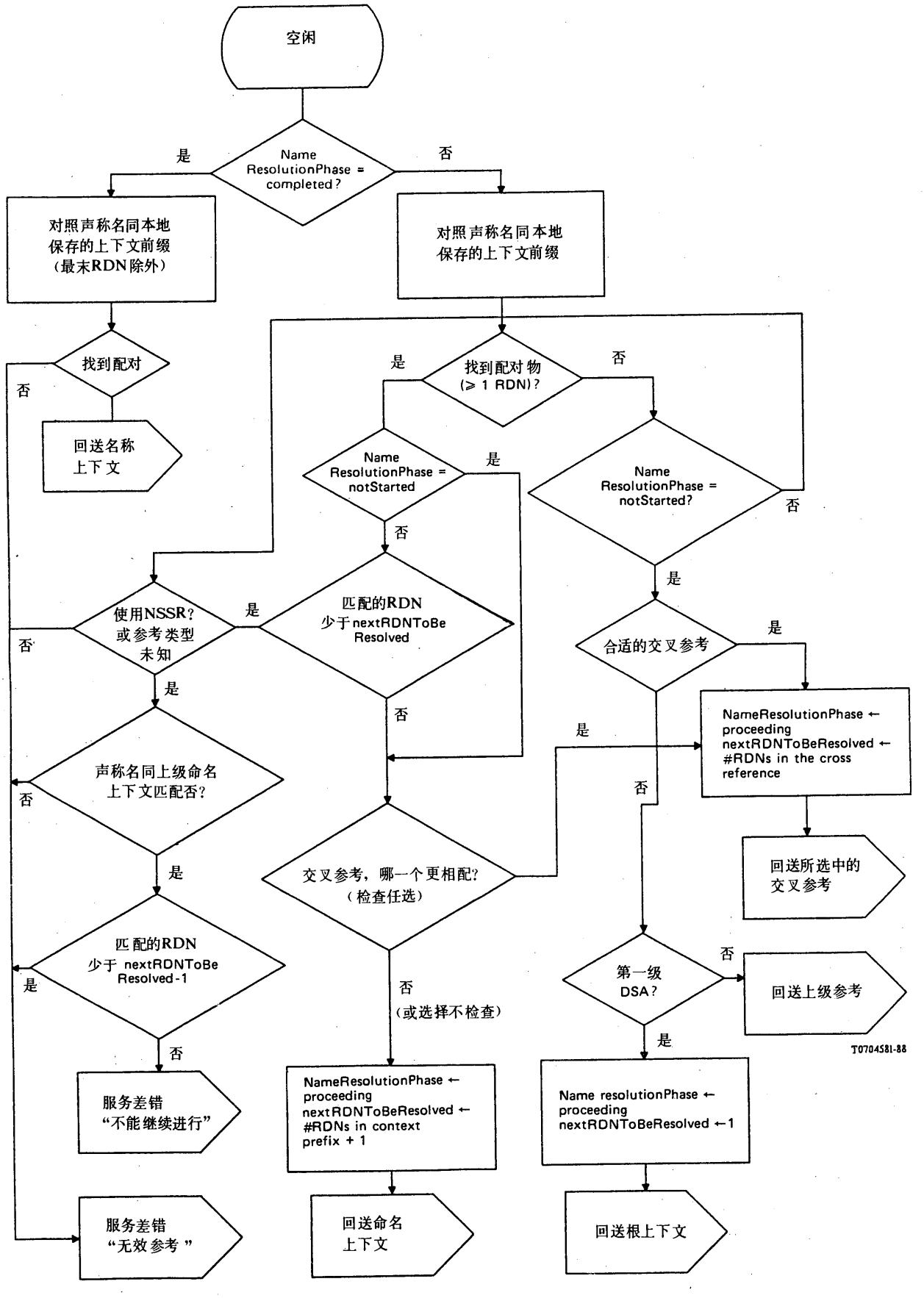


图 8/X. 518

寻找命名上下文

ARCHIVES U.I.T. GENEVE

2)本地名称鉴别(图 9/X.518)

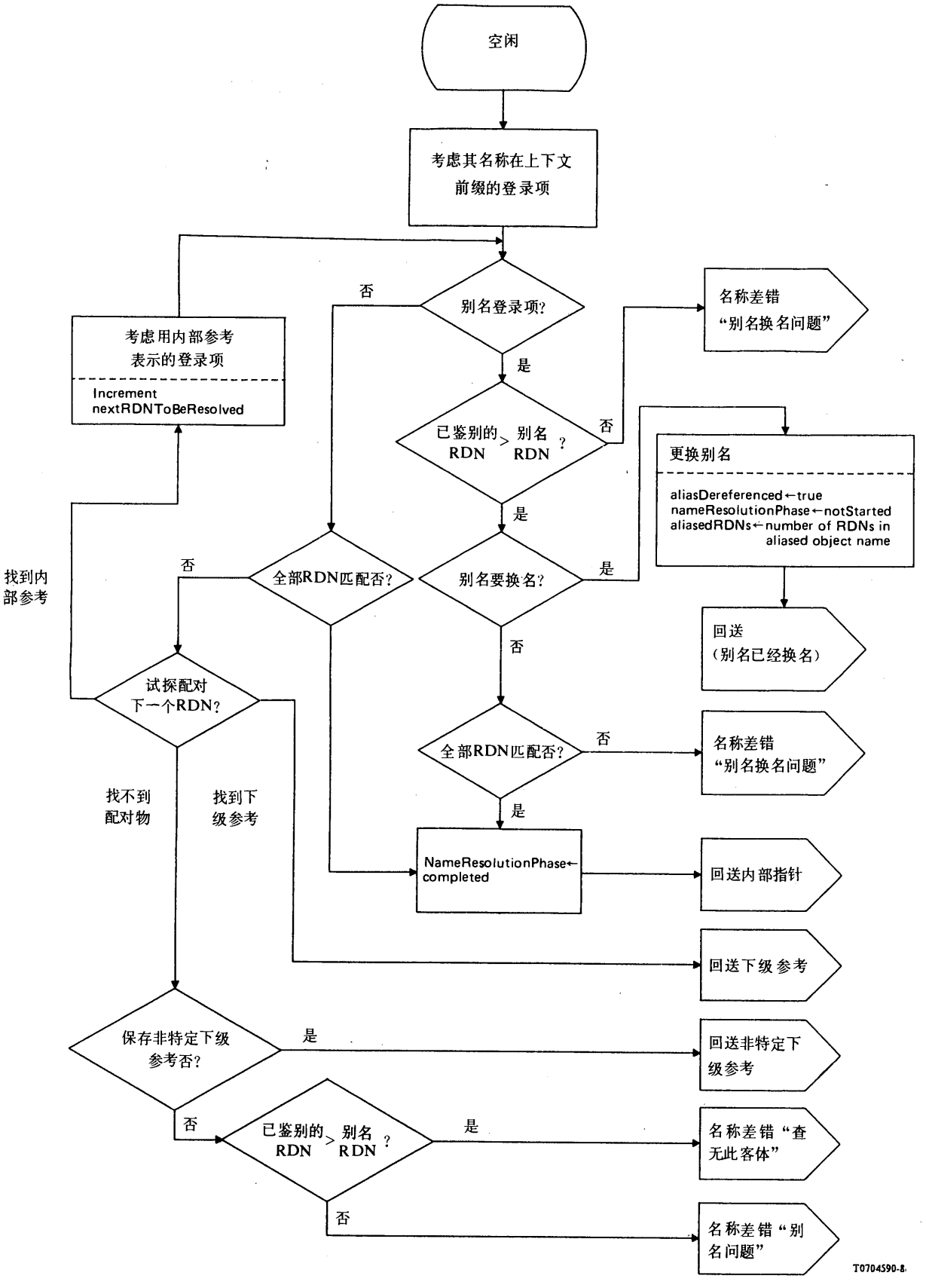


图 9/X.519

本地名称鉴别

名称鉴别规程向操作调度回送上述子规程的结果，但在下述情况下例外。其一，是当寻找命名上下文规程指出必须进一步检查的上下文并回送本地命名上下文时；其二，是当本地名称鉴别规程指出它已经更换别名时。在第一个情况下，名称鉴别规程调用本地名称鉴别规程。在第二个情况下，使用新目标客体名重新激活名称鉴别规程。

18.6.1 自变量

本规程使用下列自变量：

— 目标客体名(声称名)；
— 操作进展；
— **dontDereferenceAliases** 服务控制的值；
— **aliasedRDNs** 参数的值；
— **aliasDereferenced** 参数的值。

18.6.2 结果

成功结果有两种情况。

第一种情况回送：
— 一个参考；
— 操作进展(经过适当更新)；
— **aliasDereferenced** 指示和 **aliasedRDNs**(任选)。

第二种情况回送：
— 一个指示，表示找到了命名上下文(连同指向登录项的本地指针)；
— 操作进展(经过适当更新)；
— **aliasDereferenced** 指示和 **aliasedRDNs**(任选)。

18.6.3 差错

下列差错之一可能被回送：

— **ServiceError(unableToProceed)**；
— **ServiceError(invalidReference)**；
— **NameError (aliasProblem,noSuchObject** 或 **aliasDereferencingProblem)**。

18.6.4 规程

1) 激活寻找命名上下文规程。
2) 等待来自寻找命名上下文规程的响应。
3) 接受回送的结果或差错，即本地命名上下文找到、远端参考、无法继续操作差错、名称差错或无效参考。
4) 根据回送的结果或差错完成某些功能。
a) 倘若找到本地命名上下文，则激活本地名称鉴别规程。此规程有可能回送一个“内部参考已找到”、远端参考、别名换名或名称差错。这些结果或差错都能导致名称鉴别的中止，并报告结果。除非有一

个别名已被换名，规程从步骤 1)重新起动。

b) 其他的结果都将被回送给操作调度规程。

18.6.5 寻找命名上下文规程

18.6.5.1 绪论

图 8/X.518 表示规程流程图。下面是文字描述。在此，假设规程一旦退出，就回送操作进展当前值。

18.6.5.2 自变量

本规程使用下列自变量：

— 目标客体名(声称名)；

— 操作进展。

18.6.5.3 结果

成功的结果有两种情况。

第一种情况回送：

— 一个参考，

— 操作进展(经过适当更新)。

第二种情况回送：

— 一个指示，表示在本地找到了适当的命名上下文；

— 操作进展(经过适当更新)。

18.6.5.4 差错

有可能回送下列差错：

— 服务差错(无法继续操作)；

— 服务差错(无效参考)。

18.6.5.5 规程

1) 倘若在登录项中把 **nameResolutionPhase** 置为 **completed**，则对照声称名和在本地保存的全部命名上下文的上级命名上下文的前缀。倘若能找到匹配的一对，则回送在本地保存的全部合适的命名上下文。否则，应回送一个 **invalidReference ServiceError**。

2) 倘若未把 **nameResoLutionPhase** 置为 **completed**，则将上下文前缀同在声称名首部的一个或多个 RDN 序列进行对照。为了找到匹配对，必须从包含在一个上下文前缀的所有 RDN 中寻找。这里使用的上下文前缀是属于 DSA 管辖的命名上下文的前缀。在有多个匹配对的情况下，应选择拥有匹配 RDN 最多的一对。

   倘若找到一个匹配对，则执行步骤 3)。

   倘若找不到匹配对，则执行步骤 5)。

3) 倘若 **nameResolutionPhase** 是 **notStarted**，则执行 4)。倘若声称名首部中的 RDN 通过规程 2)获得的匹配对的总数大于或等于 **OperationProgress** 的 **nextRDNToBeResolved**，则执行步骤 4)；否则执行步骤 9)。

4） **nextRDNToBeResolved** 应置为匹配的 RDN 总数加 1，而 **nameResolutionPhase**，则应置为 **Proceeding**。然后回送上下文和结束本规程。

为了提高操作性能，DSA 可任意选择把声称名同该 DSA 保存的交叉参考对照。倘若 RDN 同交叉参考的匹配数超过本地保存的上下文前缀，则应执行步骤 7）。

注 — 在遇到这种结果的情况下，名称鉴别规程应调用本地名称鉴别规程。

5） 倘若找不到匹配对，则应检查 **nameResolutionPhase**。倘若 **nameResolutionPhase** 是 **notStarted**，则应执行步骤 6）。

倘若 **nameResolutionPhase** 是 **proceeding** 或 **completed**，则应执行步骤 9）。

6） 使用交叉参考上下文前缀，试探同声称名首部中的一个或多个 RDN 进行对照。在有多个匹配对的情况下，应选择拥有匹配 RDN 总数最多的一对。

7） 倘若找到同一个交叉参考匹配的 RDN，则应将 **nextRDNToBeResolved** 置为选择的交叉参考的 RDN 数。然后回送交叉参考并中止本规程。

8） 倘若找不到同一个交叉参考匹配的 RDN，则判定该 DSA 是否为一级 DSA。若不是，则它应当有一个上级参考。回送此参考并中止本规程。

倘若该 DSA 是一级 DSA，则应将 **nextRDNToBeResolved** 置“1”；将 **nameResolutionPhase** 置为 **proceeding**。回送根命名上下文并中止本规程。

9） 检查自变量 **ChainingArgument** 中的 **referenceType** 的值。倘若使用了非特定下级参考，或者请求是来源于 DUA，则执行步骤 10）；否则，回送差错指示 **ServiceError** 并附问题 **invalidReference**，然后中止本规程。

10） 比较声称名首部和在本地保存的命名上下文的前缀（减掉它们的最后一个 RDN）。这实质上是同本 DSA 的直接上级的某些命名上下文的比较。

倘若找不到匹配对，则应回送 **ServiceError** 附问题 **invalidReference**，然后中止本规程。

倘若找到匹配对，而且匹配 RDN 的数目少于 **nextRDNToBeResolved－1**，则应回送 **ServiceError** 附问题 **invalidReference**；否则应回送 **ServiceError** 附问题 **unableToProceed**；然后中止本规程。

18.6.6 本地名称鉴别

18.6.6.1 绪论

本地名称鉴别把声称名中的 RDN 同内部知识参考对照。本规程回送“找到”、“远端参考”、“别名已换名”或差错指示。

图 9/X.518 表示本规程的流程图，下面是文字说明。

18.6.6.2 自变量

本规程使用下列自变量：

— 指向命名上下文的内部参考（连同指向登录项的指针，该登录项的名称与上下文前缀名相同）；
— 目标客体名（声称名）；
— 操作进展；
— **dontDereferenceAliases** 服务控制值；
— **aliasedRDN** 参数值；
— **aliasDereferenced** 参数值。

18.6.6.3 结果

成功的结果共有三种情况。

第一种情况回送：

— 一个参考；

— 操作进展(经过适当更新)。

— 第二种情况回送：

— 一个指示，表示在本地找到了登录项；

— 操作进展(经过适当更新)。

第三种情况回送：

— 一个指示，表示有一个别名已换名；

— 操作进展(置为“未起动”)。

18.6.6.4 差错

可回送的差错如下：

— 名称差错。

18.6.6.5 规程

由 FindNaming 上下文回送的命名上下文应指向子树根登录项。在根上下文的情况下，其登录项只是一个零登录项。

1) 倘若内部参考指向一个别名登录项，则执行步骤 7)，否则执行步骤 2)。
2) 倘若在声称名中的全部 RDN 都匹配，则表示目标登录项已经找到。置 **nameResolutionPhase** 为 **Completed**。回送一个内部指针，然后中止本规程。
   否则必须执行步骤 3)。
   注 — 为了寻找匹配对，可单独使用上下文前缀，或使用上下文前缀加上包含在知识树上的内部参考中的 RDN 序列。
3) 倘若发现内部参考登录项在知识树上是与声称名中的下一个 RDN 匹配的当前登录项的下级，则令 **nextRDNToBeResolved** 增值，置当前登录项为下级登录项，然后再执行本规程中的步骤 1)。
4) 倘若当前的登录项有一个下级参考，其中的 RDN 同声称名的下一个 RDN 相匹配，则应回送该下级参考并中止本规程。
5) 倘若有一个非特定下级参考是知识树上的当前登录项的下级，则应回送上述非特定下级参考做为参考并中止本规程。
6) 倘若找不到内部参考、下级参考或非特定下级参考，则应检查声称名中已经匹配的 RDN 数目。倘若匹配 RDN 的数目大于 **ChainingArgument** 的 **aliasedRDN** 中匹配的 RDN 数目，则应回送 **NameError** 附问题 **noSuchObject**。倘若相匹配的 RDN 总数目少于上述数目，则应回送 **NameError** 附 **aliasProblem**。
7) 倘若在声称名中相匹配的 RDN 总数少于或等于 **ChainingArgument** 中的 **aliasedRDN**(若有的话)，则已换名的前别名(若有的话)便指向另一个别名。在此种场合，则应回送 **NameError** 附问题 **aliasDereferencingProblem**。
8) 倘若 **aliasedRDN** 消失，或相匹配的 RDN 总数大于 **ChainingArgument** 中的 **aliasedRDN**，则应检查 **dontDereferenceAlias** 服务控制。若能更换别名，则应执行步骤 9)，否则执行步骤 10)。
9) 更换别名。置 **OperationProgress** 的 **nameResolutionPhase** 为 **notStarted**。置 **ChainingArgument** 中的 **aliasDereferenced** 为 **TRUE**，并置 **aliasedRDN** 为别名登录项的 **aliasedObjectName** 属性中的 RDN 数。置 **targetObject** 为新名称。中止本规程(名称鉴别的进程将被重新起动)。
10) 倘若在声称名中的全部 **RDN** 都匹配，则执行步骤 2)。否则，应回送 **NameError** 附 **aliasDereferencingProblem**。

18.7 客体评价规程

客体评价规程包括两种规程：

a） 单客体评价规程；

b） 多客体评价规程。

图 10/X.518 表示客体评价规程。

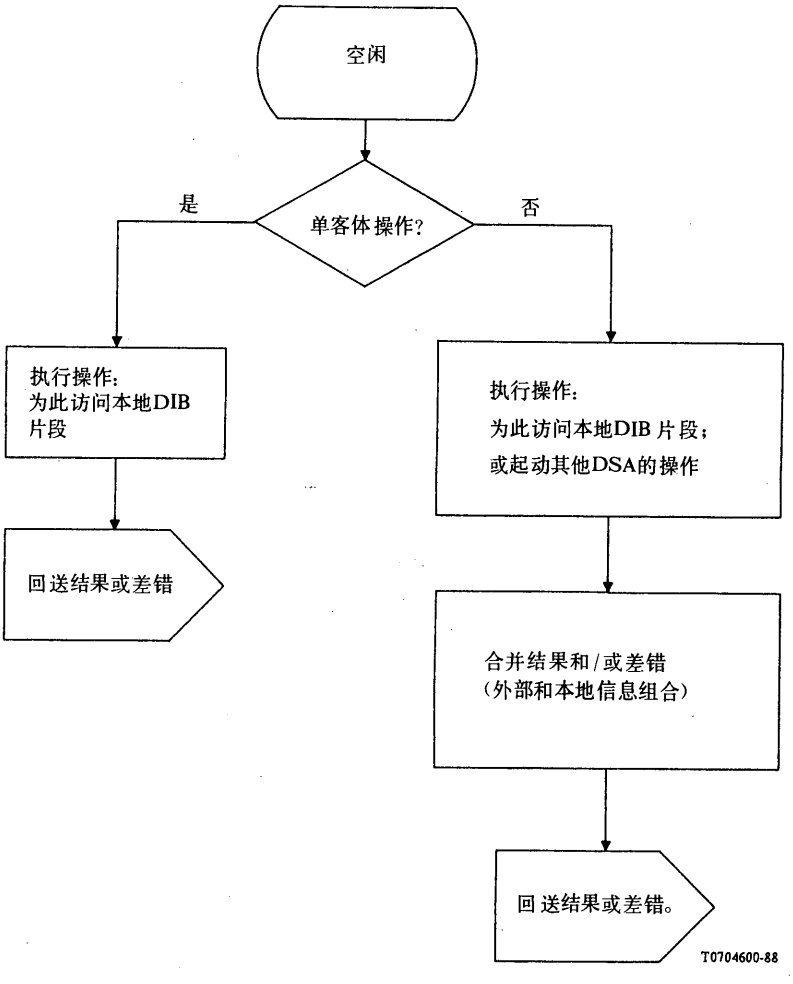


图 10/X.518

评价和结果合并规程

18.7.1 单客体评价规程

单客体评价规程适用于关心访问单客体的特定操作类别，本规程直接执行，它把结果或差错回送给起动者。

这些操作包括 **Read**，**Compare**，**AddEntry**，**RemoveEntry**，**ModifyEntry** 和 **ModifyRDN**，以及它们对应的连锁操作。

对登录项所要求的操作，在建议 X.511 的适当章节中描述。

AddEntry,**RemoveEntry** 和 **ModifyRDN** 操作均影响知识。倘若登录项的直接上级是在另外一个 DSA，则应保持正确的外部知识参考。至于如何保持则不属于本建议的讨论范围。

如何选择 DSA，使它包含由 **AddEntry** 产生的登录项的问题，亦不属于本建议的讨论范围。

倘若由 **AddEntry** 产生的、或由 **ModifyRDN** 修改的一个登录项的直接上级拥有一些非特定下级参考，则应遵照本建议以外的规程确保两个登录项中的可识别名彼此不相同。

在这些条件下，无法满足的一些请求都将以 **UpdateError** 附问题 **affectsMultipleDSA** 宣告失败。

18.7.2 多客体评价规程

多客体评价规程适用关心访问多客体的特定操作类别，这些规程在下列章节中描述。

这些操作包括 **List** 和 **Search**，以及对应的连锁操作。

18.7.2.1 列表

本节详述 **List** 和 **ChainedList** 的评价规程(下文中的"**List**"包括上述两种情况)。

18.7.2.1.1 列表规程(Ⅰ)

本规程适用于列表请求包含的 **OperationProgress** 中的 **nameResolutionPhase** 置为 **notStarted** 或 **proceeding**，以及 DSA 执行名称鉴别规程后发现它保存基础客体的场合。

基础客体将用符号"e"表示。

1) 取本地保存的 e 的直接下级，组成本地结果集。把 **ListResult** 中的 **aliasEntry** 和 **fromEntry** 置为适当的值。
2) 取保存"e"的直接下级的 DSA 的非特定下级参考集和下级参考。
3) 把包含基础客体＝e 和 **Operationprogress** 置为 **Completed** 的子请求传给操作调度，后者又把该子请求传给保存 e 的直接下级的每一个 DSA。

注 — 倘若 DSA 保存一些下级参考，后者包含表示该下级登录项是否为别名的指示，而且 **dontUseCopy** 为 **FALSE**，则对这些登录项而言本步骤可以省略。有关下级的信息可以直接获得。

18.7.2.1.2 列表规程(Ⅱ)

本规程适用于列表请求，即当该请求中的 **Operationprogress** 所属的 **nameResolutionPhase** 置为 **Completed** 时的场合。

基础客体将用符号"e"表示。

1) 利用本地保存的 e 所属的每一个直接下级组成一个本地结果集。置 **ListResult** 中的 **aliasEntry** 和 **fromEntry** 为适当值。
2) 把结果交给操作调度，然后由后者把它们传给请求者 DUA 或 DSA。

18.7.2.2 搜索

本节详述 **Search** 和 **ChainedSearch** 的评价规程(下文中的"搜索"包括上述两种方式)。

请注意，这里有两种情况需要分别采用两种规程。第一种规程(§18.7.2.2.1)适用于当执行搜索的 DSA 包含一个本地登录项 **targetObject** 时的情况。第二种规程(§18.7.2.2.2)适用于当执行搜索的 DSA 不包含 **targetObject** 而只包含 **targetObject** 的下级时的情况。

18.7.2.2.1 搜索规程(Ⅰ)

本规程适用于搜索请求,即当搜索请求中的 **OperationProgress** 所属的 **nameResolutionPhase** 置为 **notstarted** 或 **proceeding** 时,以及当 DSA 执行名称鉴别之后决定保存目标客体时的情况。

基础客体将以符号"e"表示。

1) 倘若自变量 **Subset** 为 **baseObject** 或 **wholeSubtree**,则将搜索指定的过滤器自变量应用于登录项 e,以组成本地结果集。为结果的合并回送结果。倘若自变量 **subset** 为 **baseObject**,则中止本规程;否则从步骤 2)继续执行。

2) 倘若自变量 **subset** 为 **oneLevel** 或 **wholeSubtree**,则使用本地保存的 e 的直接下级组成一个 E 集合,下述情况除外:

   倘若要更换别名,即参数 **searchAliases** 为 **TRUE**,则对所有找到的别名登录项应按 5)进行处理,而不应当把它们加到这些结果中。

   把过滤器自变量应用于 E,组成一个过滤子集 E′gE;把这个本地结果集 E′ 回送给请求者以便进行结果合并。

3) e 的其他下级有可能驻留在其他的 DSA,果真如此,它们将以下级或非特定下级参考表示。对这些参考表示的每一个 DSA 应准备一个新的搜索,其中的 targetObject=e,并把 **OperationProgress** 中的 **nameResolutionPhase** 置为 **completed**。把每一个搜索子请求回送给操作调度,让后者予以传递。倘若从一个子请求那里获得一个差错结果,则不予理睬,仿佛未发送过任何子请求。

4) 倘若自变量 **subset** 为 **oneLevel**,则表示搜索已经结束,故应中止本规程。

   倘若自变量 **subset** 为 **wholeSubtree**,则:

   倘若 2)中的 E 集合为空,则本 DSA 保存的整个子树已经被搜索完毕,故应中止本规程;

   否则遵照下列方法继续处理:

   让 E 集合中的每一个登录项都用符号 e 表示。然后从 2)开始对每一个登录项 e 重复执行搜索规程。

5) 倘若拟对别名进行更换,则应把在步骤 2)找到的任何别名登录项都放入 D 集合中。对于在 D 中的每一个登录项 d,应把其中的别名加以更换,然后组织一个新的搜索:把 **nameResolutionPhase** 置为 **notStarted**,并使用由 **aliasedObjectName** 属性和旧的 **targetobject** 名建立的 **targetObject**。

   倘若自变量 **subset** 为 **oneLevel**,则应在新的子请求中置之为 **baeObject**,否则置之为 **wholeSubtree**。

   倘若由子请求那里回送差错结果,则不予理睬,仿佛未提出过任何子请求。

18.7.2.2.2 搜索规程(Ⅱ)

本规程适用于搜索请求,即当其中的 **OperationProgress** 所属的 **nameResolutionPhase** 置为 **completed** 时的场合。

目标客体将以符号"e"表示。

对本地保存的 e 的每一个直接下级 e′,组织一个新的请求:令 targetObject=e′。倘若自变量 **Subset** 为 **oneLevel**,则置之为 **baeObiect**,否则让它等于 **wholeSubtree**。然后,执行 §18.7.2.2.1 中的步骤 1)至 5)的程序。倘若没有那样的下级,则回送 **unableToProceed ServiceError**。

18.8 结果合并规程

本规程是当产生外部结果和/或差错(也可能是一个内部结果)时被调用的。假设所有的结果和差错都保存在 DSA 内直至本规程结束为止。

外部信息可能是由连锁操作、多点传播或请求分解产生的。

在连锁操作的情况下，可能有一个结果或差错。在多点传播的情况下，可能有一个结果、多个结果或没有结果，另外还会有一些差错。假设有一个以上的结果，则除了其中的一个结果之外，其余的结果都可任意放弃。总是希望能够回送一个结果，而不是差错指示。倘若没有结果，则应回送一个差错，但下列情况例外：

i) 倘若回送的是 **invalidReference**，则照此记录该参考，DSA 可以使用适当的外部参考继续执行该请求，或向请求者回送 **ditError**（对无效外部参考的处理方法不属于本建议的讨论范围。）

ii) 在多点传播的情况下，应不理睬 **unableToProceed** 差错，除非所有的响应都属于这个类型；在后一种情况下应向响应者回送 **NameError noSuchObject**。倘若至少有一个结果被回送，则所有的差错都可不予理睬。

iii) 在使用转移参考的情况下，这些都可以不当差错处理，而且可对它进行操作。

倘若由于请求分解而需要进行合并，则这相当于对结果进行逻辑相加。

在分解的情况下，即需要把结果和差错分别进行合并的情况下，则应将不完整的结果回送给请求者。

在这个阶段，DSA 有可能选择从送来的结果和差错中提取转移参考（该结果和差错是准备合并的）。从而 DSA 有可能决定进一步调查上述结果或差错的全部或一部分。在此情况下，采用的是连锁操作。旧的结果应加以保存，然后把它们同连锁操作产生的结果或差错合并在一起。

对于在回送的结果中可能存在的签名的处理在 §18. 9. 2 中详述。

18. 9 分布式验证规程

本节详述支持号码簿分布式验证服务所需要的规程。这些服务和有关的规程分类如下：

— 发起者验证：提供无保护（凭简单身份证）和有保护（凭数字签名）两种形式的支持；

— 结果验证：提供有保护的支持（凭数字签名）。

18. 9. 1 发起者验证

18. 9. 1. 1 凭身份证的验证

凭身份证的验证服务允许各 DSA 对原请求者的身份进行验证以达到实现本地访问控制的目的。拟利用本服务的 DSA 必须使用下列规程：

— 要求对 DAP 请求进行验证的 DSA，可在建立 DUA（DUA 或 DSA）关联的时刻通过耦合规程获得请求者的可识别名。成功地执行这些规程丝毫也不降低因利用上述关联处理一些操作可能需要的验证水平；

— 同 DUA 有关联的 DSA，为了同其他 DSA 进行连锁操作，必须将请求者的可识别名放入 **ChainingArgument** 中的发起者字段；

— 接受连锁操作的 DSA 在试图满足操作要求时可能成功，也可能失败，这就取决于访问权的确定结果（在本地定义的一个机制）。倘若结果不能令人满意，则可回送一个 **SecurityError**，并把 **SecurityProblem** 置为 **insufficientAccessRights**。

18. 9. 1. 2 凭签字的发起者验证

凭签字的发起者验证服务允许 DSA 对特定服务请求的发起者（以有保护的方式）进行验证。DSA 为实现本服务应执行的规程在本节中描述。

凭签字的验证服务是由 DUA 使用可选签字服务请求中的 **SIGNED** 服务请求调用的。

当一个 DSA 由另一个 DSA 那里收到一个签署请求时，该 DSA 应在着手处理该操作之前先把另一个 DSA 的签字删除。假设验证签字的结果令人满意，该 DSA 就可以继续操作。倘若在处理的过程中 DSA 要

求使用连锁方式、多点传播或请求分解，则每一个连锁操作使用的自变量集应构造如下：

— DSA 组织一个自变量集，是否要署名可以任意选择；自变量集包括输入的署名自变量集和一个经修改的 **ChainingArgument**。

在 DSA 能够给响应提供信息的场合，为了确定发起者对该信息的访问权，DSA 可以使用凭署名服务请求对该发起者进行验证。

倘若 DSA 收到一个不署名的服务请求，要求提供信息，但由于信息是凭对发起者的验证发放的，故应回送一个 **SecurityError** 并把 **Security Problem** 置为 **protectionRequired**。

18. 9. 2 结果验证

提供本服务的目的是为了使号码簿操作请求者（DUA 或 DSA）能够对结果的来源进行检验（采用数字签名技术以保证安全可靠）。申请本结果验证服务可不考虑是否使用了发起者验证。

结果验证服务是使用 **protectionRequest** 的 signed 值起动的，该 **protectionRequest** 是号码簿操作的自变量集内的一个成份；接受选择这个任选服务操作的 DSA，就可以任意选择签署即将产生的任何结果。在 **protectionRequest** 中的 **signed** 任选项向 DSA 指出请求者的选择；DSA 可能签署，但也可能不签署任何即将产生的结果。

在 DSA 执行连锁操作、多点传播或分解这样的请求的情况下，DSA 就拟回送给请求者结果的形式而言有很多选择，即：

a) 向请求者回送一个混合响应（署名或不署名的）；

b) 向请求者回送由两个或多个未经逻辑相加的部分响应组成的一个集合（署名或不署名的）；在此集合内，有零个或多个成员可以是署名的，另外有零个或一个成员是不署名的。在有一个不署名的部分结果的情况下，此成员实际上有可能是由其他的 DSA 收到的、由此 DSA 提供的，或由其他的 DSA，和此 DSA 提供的一个或多个不署名的部分响应的逻辑和。

附 件 A

（附于建议 X. 518）

分布式操作使用的 ASN. 1

本附件是本建议的一部分。

本附件包括在本建议中出现的所有 **ASN**. 1 类型、值和宏定义，并以 **ASN**. 1 模块 **Distributed Operation** 的形式提供。

```
DistributedOperations {joint-iso-ccitt  ds(5)  modules(1)  distributedOperations(3)}
DEFINITIONS  ::=
BEGIN
```

```
EXPORTS
    DirectoryRefinement, chainedReadPort, chainedSearchPort, chainedModifyPort,
    DSABind, DSABindArgument,
    DSAUnbind,
    ChainedRead, ChainedCompare, ChainedAbandon,
    ChainedList, ChainedSearch,
    ChainedAddEntry, ChainedRemoveEntry,
    ChainedModifyEntry, ChainedModifyRDN,
    DsaReferral, ContinuationReference;

IMPORTS
    InformationFramework, abstractService, distributedOperations,
    directoryObjectIdentifiers, selectedAttributeTypes
        FROM    UsefulDefinitions {joint-iso-ccitt ds(5) modules(1) usefulDefinitions(0)}

    DistinguishedName, Name, RelativeDistinguishedName
        FROM    InformationFramework informationFramework

    id-ot-dsa, id-pt-chained-read, id-pt-chained-search, id-pt-chained-modify
        FROM    DistributedDirectoryObjectIdentifiers, distributedDirectoryObjectIdentifiers

    PresentationAddress
        FROM    SelectedAttributeTypes selectedAttributeTypes

    directory, readPort, searchPort, modifyPort
    DirectoryBind,
    ReadArgument, ReadResult,
    CompareArgument, CompareResult,
    Abandon
    ListArgument, ListResult,
    SearchArgument, SearchResult,
    AddEntryArgument, AddEntryResult,
    RemoveEntryArgument, RemoveEntryResult,
    ModifyEntryArgument, ModifyEntryResult,
    ModifyRDNArgument, ModifyRDNResult,
    Abandoned, AttributeError, NameError, ServiceError, SecurityError, UpdateError
    OPTIONALLY-SIGNED, SecurityParameters
        FROM    DirectoryAbstractService directoryAbstractService

-- objects and ports --

DirectoryRefinement ::= REFINE directory AS
    dsa     RECURRING
        readPort            [S]     VISIBLE
        searchPort          [S]     VISIBLE
        modifyPort          [S]     VISIBLE
        chainedReadPort             PAIRED WITH dsa
        chainedSearchPort           PAIRED WITH dsa
        chainedModifyPort           PAIRED WITH dsa

dsa OBJECT
        PORTS { readPort        [S],
                searchPort      [S],
                modifyPort      [S],
                chainedReadPort,
                chainedSearchPort
                chainedModifyPort}
    ::= id-ot-dsa

chainedReadPort PORT
    ABSTRACT OPERATIONS {
        ChainedRead, ChainedCompare,
        ChainedAbandon}
    ::= id-pt-chained-read

chainedSearchPort PORT
    ABSTRACT OPERATIONS {
        ChainedList, ChainedSearch}
    ::= id-pt-chained-search
```

```
chainedModifyPort PORT
    ABSTRACT-OPERATIONS {
        ChainedAddEntry, ChainedRemoveEntry,
        ChainedModifyEntry, ChainedModifyRDN}
    ::=  id-pt-chained-modify

DSABind  ::=  ABSTRACT-BIND
    TO  {chainedRead,
        chainedSearch,
        chainedModify}
    DirectoryBind

DSAUnbind::=  UNBIND
    FROM        {chainedRead,
                chainedSearch,
                chainedModify}

-- operations, arguments and results --

ChainedRead  ::=
    ABSTRACT-OPERATION
        ARGUMENT        OPTIONALLY-SIGNED   SET{
                                            ChainingArgument,
                                            [0] ReadArgument}
        RESULT          OPTIONALLY-SIGNED   SET{
                                            ChainingResult,
                                            [0] ReadResult}
        ERRORS {
            DsaReferral, Abandoned, AttributeError, NameError,
            ServiceError, SecurityError}

ChainedCompare  ::=
    ABSTRACT-OPERATION
        ARGUMENT        OPTIONALLY-SIGNED   SET{
                                            ChainingArgument,
                                            [0] CompareArgument}
        RESULT          OPTIONALLY-SIGNED   SET{
                                            ChainingResult,
                                            [0] CompareResult}
        ERRORS {
            DsaReferral, Abandoned, AttributeError, NameError,
            ServiceError, SecurityError}

ChainedAbandon  ::=  Abandon

ChainedList  ::=
    ABSTRACT-OPERATION
        ARGUMENT        OPTIONALLY-SIGNED   SET{
                                            ChainingArgument,
                                            [0] ListArgument}
        RESULT          OPTIONALLY-SIGNED   SET{
                                            ChainingResult,
                                            [0] ListResult}
        ERRORS {
            DsaReferral, Abandoned, AttributeError, NameError,
            ServiceError, SecurityError }

ChainedSearch  ::=
    ABSTRACT-OPERATION
        ARGUMENT        OPTIONALLY-SIGNED   SET{
                                            ChainingArgument,
                                            [0] SearchArgument}
        RESULT          OPTIONALLY-SIGNED   SET{
                                            ChainingResult,
                                            [0] SearchResult}
        ERRORS {
            DsaReferral, Abandoned, AttributeError, NameError,
            ServiceError, SecurityError}
```

```
ChainedAddEntry ::=
    ABSTRACT-OPERATION
        ARGUMENT        OPTIONALLY-SIGNED    SET{
                                             ChainingArgument,
                                             [0] AddEntryArgument}
        RESULT          OPTIONALLY-SIGNED    SET{
                                             ChainingResult,
                                             [0] AddEntryResult}
        ERRORS {
            DsaReferral, Abandoned, AttributeError, NameError,
            ServiceError, SecurityError, UpdateError}

ChainedRemoveEntry ::=
    ABSTRACT-OPERATION
        ARGUMENT        OPTIONALLY-SIGNED    SET{
                                             ChainingArgument,
                                             [0] RemoveEntryArgument}
        RESULT          OPTIONALLY-SIGNED    SET{
                                             ChainingResult,
                                             [0] RemoveEntryResult}
        ERRORS {
            DsaReferral, Abandoned, NameError,
            ServiceError, SecurityError, UpdateError}

ChainedModifyEntry ::=
    ABSTRACT-OPERATION
        ARGUMENT        OPTIONALLY-SIGNED    SET{
                                             ChainingArgument,
                                             [0] ModifyEntryArgument}
        RESULT          OPTIONALLY-SIGNED    SET{
                                             ChainingResult,
                                             [0] ModifyEntryResult}
        ERRORS {
            DsaReferral, Abandoned, AttributeError, NameError,
            ServiceError, SecurityError, UpdateError}

ChainedModifyRDN ::=
    ABSTRACT-OPERATION
        ARGUMENT        OPTIONALLY-SIGNED    SET{
                                             ChainingArgument,
                                             [0] ModifyRDNArgument}
        RESULT          OPTIONALLY-SIGNED    SET{
                                             ChainingResult,
                                             [0] ModifyRDNResult}
        ERRORS {
            DsaReferral, Abandoned, NameError,
            ServiceError, SecurityError, UpdateError}

-- errors and parameters --

DSAReferral ::=
    ABSTRACT-ERROR
        PARAMETER SET {
            [0] ContinuationReference,
            contextPrefix [1] DistinguishedName OPTIONAL}

-- common arguments/results --

ChainingArguments ::=       SET {
    originator              [0]  DistinguishedName OPTIONAL,
    targetObject            [1]  DistinguishedName OPTIONAL,
    operationProgress       [2]  OperationProgress DEFAULT {notStarted}
    traceInformation        [3]  TraceInformation,
    aliasDereferenced       [4]  BOOLEAN DEFAULT FALSE,
    aliasedRDNs             [5]  INTEGER OPTIONAL,
        -- absent unless aliasDereferenced is TRUE
```

```
        returnCrossRefs         [6]     BOOLEAN DEFAULT FALSE,
        referenceType           [7]     ReferenceType DEFAULT superior,
        info                    [8]     Domaininfo OPTIONAL,
        timeLimit               [9]     UTCTime OPTIONAL,
                                [10]    SecurityParameters DEFAULT { }}

ChainingResults         ::=     SET {
        info                    [0]     Domaininfo OPTIONAL,
        crossReferences         [1]     SEQUENCE OF CrossReference OPTIONAL,
                                [2]     SecurityParameters DEFAULT { }}

CrossReference          ::=     SET {
        contextPrefix   [0]     DistinguishedName,
        accessPoint             [1]     AccessPoint}

ReferenceType           ::=     ENUMERATED {
                superior                        (1),
                subordinate                     (2),
                cross                           (3),
                nonSpecificSubordinate          (4)}

TraceInformation        ::=     SEQUENCE OF
        SEQUENCE {
                targetObject    Name,
                dsa Name,
                OperationProgress}

OperationProgress       ::=     SET     {
        nameResolutionPhase             [0]  ENUMERATED {
                                                notStarted  (1),
                                                proceeding  (2),
                                                completed   (3)},
        nextRDNToBeResolved             [1]  INTEGER OPTIONAL}

DomainInfo      ::=     ANY

ContinuationReference           ::=     SET {
        targetObject            [0]     Name,
        aliasedRDNs             [1]     INTEGER OPTIONAL,
        operationProgress       [2]     OperationProgress
        rdnsResolved            [3]     INTEGER OPTIONAL,
        referenceType           [4]     ReferenceType OPTIONAL,
                                -- only present in the DSP --
        accessPoints            [5]     SET OF AccessPoint }

AccessPoint     ::=     SET {
        ae-title        [0]     Name,
        address         [1]     PresentationAddress }
```

附 件 B

（附于建议 X. 518）

知 识 模 型

本附件不是本建议的一部分。

B. 1 知识模型的范例

下面的范例说明图 5/X. 518（§9）所示的 DSA 要保存的知识信息。图 5/X. 518 绘出一个假想的 DIT，图中将 DIT 在逻辑上分成五个命名上下文（A，B，C，D 和 E），并在物理分布于三个 DSA（DSA1，DSA2，

DSA3)。其中，DSA1 存放上下文 C，DSA2 存放上下文 A，B 和 E，而 DSA3 存放上下文 D。

图 B － 1/X. 518 至 B － 3/X. 518 使用下列缩写词：

SUPR：　上级参考
SUBR：　下级参考
INTR：　内部参考
NSSR：　非特定下级参考
CROSSR：　交叉参考
DSAn：　DSAn 的可识别名
PS：　表示地址
CP：　上下文前缀
RDN：　相对可识别名
DSA：　DSA 的可识别名
PTR：　指针
AON：　别名客体名

注 — 下面的示意图只是为了在本节定义的一些概念提供图解说明有关有特定 DSA 实现中实际储存和管理知识信息的方法属于本地决策而不属于本建议的讨论范围。

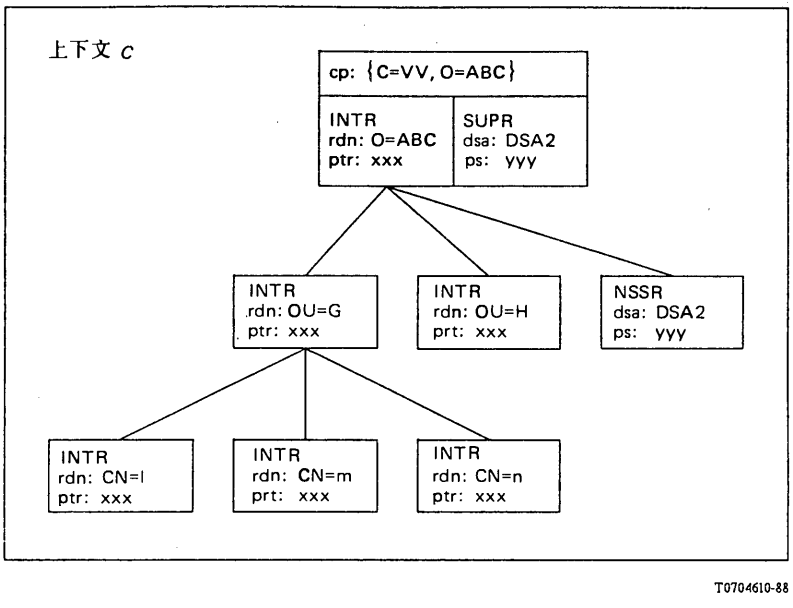


图 B - 1/X. 518

DSA1 的知识信息

图 B - 1/X. 518 说明 DSA1 必须保存的知识信息。该知识信息必须包括下列上下文前缀和一组参考：

上下文前缀：　{C＝WW，O＝ABC}，上下文 C.
交叉参考：　{}
上级参考：　{DSA2，DSA2 的表示地址}
上下文 C 的内部参考：　{C＝WW，O＝ABC}，
　{OU＝G}，{OU＝H}
　{OU＝G，CN＝1}，
　{OU＝G，CN＝m}，

{OU=G，CN=n}。

下级参考：　　{　}

非特定下级参考：　　{DSA2，DSA2 的表示地址}。

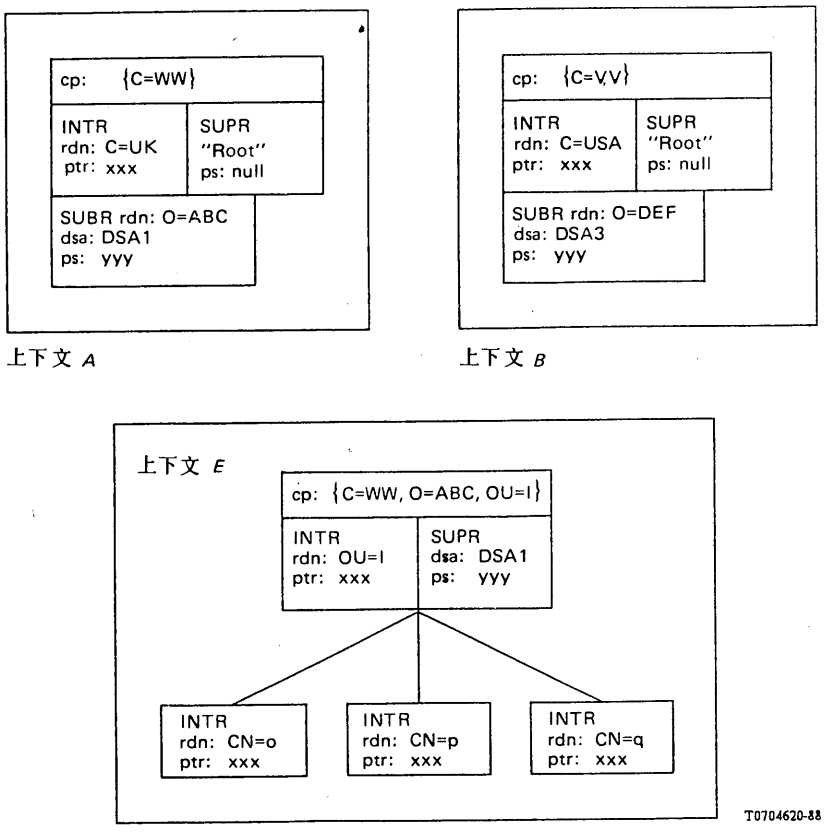


图 B － 2/X. 518

DSA2 的知识信息

图 B－2/X. 518 说明 DSA2 必须保存的知识信息。该知识信息必须包括下列上下文前缀和一组参考：

上下文前缀：　　{C=WW}，上下文 A
{C=VV}，上下文 B
{C=WW}，O=ABC，OU=I}，
上下文 E。

交叉参考：　　{　}

上级参考：　　{　}

上下文 A 的内部参考：　　{C=WW}

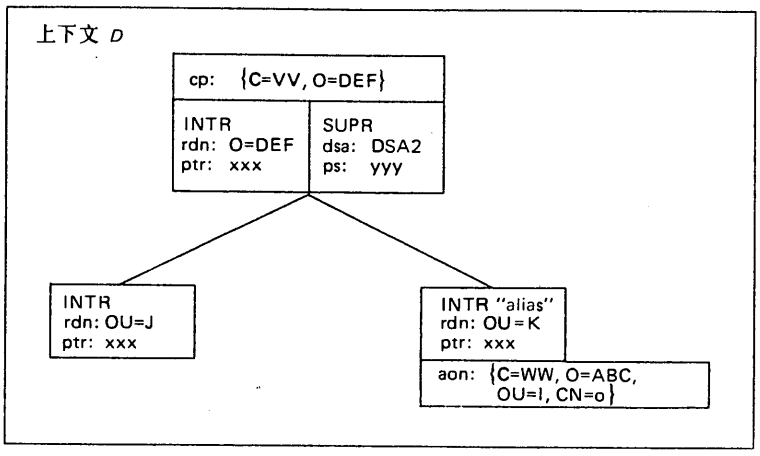
上下文 B 的内部参考：　　{C=VV}

上下文 E 的内部参考：　　{C=WW，O=ABC，OU=I}
{CN=o}，
{CN=p}，
{CN=q}。

上下文 A 的下级参考：　　{C=WW，O=ABC}

上下文 B 的下级参考：　　{C=VV，O=DEF}

非特定下级参考：　　{　}



T0704630-88

图 B-3/X. 518

DSA3 的知识信息

图 B-3/X. 518 说明 DSA3 必须保存的知识信息。该知识信息必须包括下列上下文前缀和一组参考：

上下文前缀： {C=VV，O=DEF}，上下文 D
交叉参考： { {C=WW，O=ABC，OU=H}，DSA1，
DSA1 的表示地址}（上图未指出）
上级参考： {DSA2，DSA2 的表示地址}
上下文 D 的内部参考：
{DSA1，DSA1 的表示地址}
{C=VV，O=DEF}，
{OU=J}，
{OU=K} 是 {C=WW，O=ABC，OU=I，CN=O} 的别名（别名信息不是知识信息的一部分）
下级参考： { }
非特定下级参考： { }

B. 2 分布式名称鉴别的范例

下面是关于如何使用分布式名称鉴别去处理不同的号码簿请求的范例。本范例是依据图 5/X. 518（§9）所示的假想 DIT 以及图 B-1/X. 518 至 B-3/X. 518（附件 B）所示的相应 DSA 布局。

假设采用的是一个连锁传播方式，则对 DSA1 提出的下列请求应处理如下：

1） 使用可识别名 {C=WW，O=ABC，OU=G，CN=l} 的请求
— 同上下文 C 的上下文前缀 {C=WW，O=ABC} 相匹配，DSA1 对上下文 C 拥有行政管理权。因此，名称鉴别应从 DSA1 内的上下文 C 开始执行。
名称鉴别应在上下文 C 内往下进行，成功地找出其余的 RDN，直至找到 CN=l 为止。

2） 使用可识别名 {C=WW，O=JPR} 的请求
— 不能同 DSA1 保存的任何上下文前缀相匹配，故 DSA1 应使用它的上级参考把该请求转移到

它的上级 DSA，即 DSA2。

— 在 DSA2 中，该请求同上下文前缀 {C=WW} 相匹配，故名称鉴别应在 DSA2 内的上下文 A 中开始进行。

— 名称鉴别不能在 C=WW 的下面找到同 RDNO=JPR 相匹配的下级，故该请求将以失败而告终，并确定该名称为无效（即寻找一个不存在的客体）。

3） 使用可识别名 {C=VV，O=DEF，OU=K} 的请求

— 不能同 DSA1 保存的任何上下文前缀相匹配。

— 因此，DSA1 应把该请求转移到它的上级 DSA，即 DSA2。

— 该请求同 DSA2 所保存的上下文 B 的上下文前缀 {C=VV} 相匹配。因此，名称鉴别应在 DSA2 内从上下文 B 开始进行。

— 当名称鉴别试图寻找 O=DEF 进行匹配时，它会找到一个下级参考，指出 {C=VV，O=DEF} 是保存在 DSA3 内的一个新上下文的首部。

— 名称鉴别将在 DSA3 内继续进行，直至找到 {C=VV，O=DEF，CN=K} 为止。

— 假设需要更换一些别名，则新的名称应使用包含在登录项内的别名 {C=VV，O=DEF，CN=K} 构成。新的名称为：{C=WW，O=ABC，OU=I，CN=O}。

— DSA3 将使用更换别名之后获得的新名称恢复操作。

附 件 C

（附于建议 X. 518）

验证的分布式应用

本附件不是本建议的一部分。

C. 1 摘要

保密模型在建议 X. 501 § 10 中定义。下面是该模型的重点摘要。

a） 在 DSP 中不支持操作发起者的简单验证。

b） 在 DSP 中支持强验证，即采用署名的请求和结果的验证。

c） 在 DSP 中不支持请求或结果的译码。

d） 在 DSP 中不支持对差错的验证，包括对转移参考的验证。

本附件描述如何在分布式号码簿里实现 b）。本附件使用建议 X. 509 定义的名词术语和符号。

C. 2 简单验证

对 DUA 的验证是做为 DAP 耦合操作的一部分进行的。然后，在 DSP 中的连锁自变量的发起者字段只需携带 DUA 的名称即可。

C.3 分布式验证模型

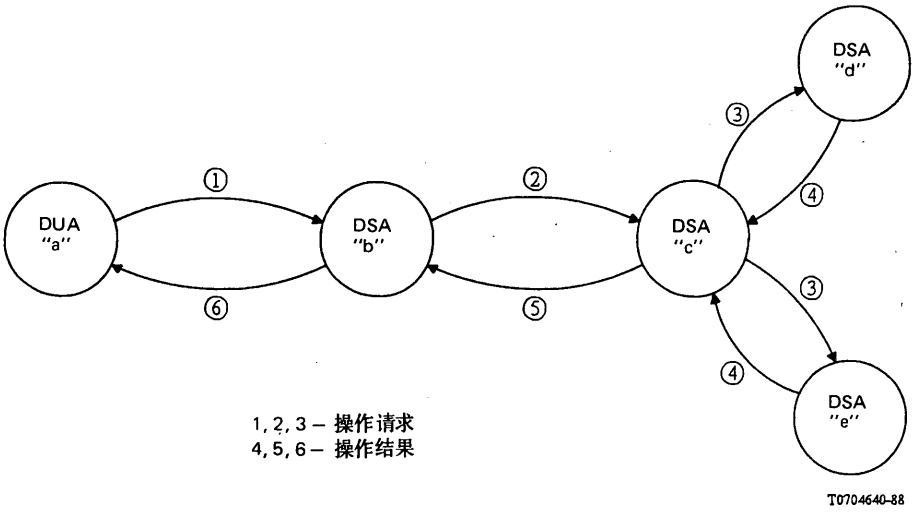


图 C-1/X.518

分布式验证模型

图 C-1/X.518 表示描述分布式验证规程时要用到的模型。该模型指出在一般情况下列表操作和搜索操作中的信息流序列。设操作是从 DUA"a"开始的,并指出目标客体是在 DSA"c"之中;在操作的过程中,DSA"b"、"c"、"d"和"e"都将被涉及。

DUA"a"首先与某一个 DSA(DSA"b")联系,后者不包含目标客体但能通过连锁方式转移到包含目标客体的 DSA(DSA"c")。假设所有的 DSA 都以转移参考方式进行操作,则上述模型可大大简化;每一个 DUA/DSA 的交换,在验证术语里,相当于 DUA"a"和 DSA"b"之间的交互式操作。

C.4 DUA 至 DSA

发起者的验证是通过交换(1)的结果实现的。在图 C-1/X.518 中,验证的规程如下:

令

OA=操作自变量,即搜索、读、比较等在第 3 部分定义的自变量。

和

a(OA)=由 DUA"a"签署的操作自变量。

验证是通过检验签名确定的。

C.5 由 DAP 向 DSP 的转移

本规程由图 C-1/X.518 所示的 DSA"b"执行,表示发起者的署名身份证由 DAP 向 DSP 的转移。

DSA"b"组成如本建议 § 12.3 所述的适当连锁自变量,并把它同 DAP 的操作自变量合并起来,构成 DSP 的一个连锁操作,即连锁读、搜索、列表等。在把如此构成的连锁操作传递给其他 DSA(如图 C-1/X.518 中的 DSA"c")之前必须先加以签署。相应的数据结构可表示如下:

b{ChA,a{OA}}=由 DSA b 签署的连锁操作

其中

ChA=连锁自变量。

因此,在 DSP 中[图 C-1/X.518 中标以交换②],DSA 之间传递的验证信息包括两个部分:

— 由发起者署名的操作自变量,它允许对发起者进行验证;

— 由发送者 DSA 签署的连锁操作，它允许对发送者 DSA 进行验证。

C.6 通过中间 DSA 的连锁操作

本规程将由图 C-1/X.518 所示的 DSA“c”执行。DSA“c”将把由发送者 DSA（图 C-1/X.518 中的 DSA“b”）提供的签名放弃，并修改本建议 § 12.3 所述的连锁自变量。然后，DSA“c”把已经修改过的连锁自变量同署名的操作自变量合并，最后在结果上签名建立一个修改过的署名连锁操作。本规程可描述如下：

c{ChA′a{OA}}＝由 DSA，“c”签署的连锁操作。

其中，

ChA′＝修改过的连锁自变量。

修改过的连锁操作在图 C - 1/X.518 中是用交换③表示的。取决于操作的性质和所保存的知识类型，DSA“c”可在对任何合成操作进行连锁传递或多点传播之前先完成对请求的分解操作。这个动作在图 C-1/X.518 中是这样表示的：DSA“c”把操作发送到 DSA“d”和 DSA“e”；在每一种情况下，使用的验证规程是一样的。

C.7 结果验证

号码簿操作发起者使用 **protectionRequest SecurityParameter** 内的 **signed** 任选项请求结果验证服务。在响应上述请求的过程中，DSA 可任意选择是否要签署某一个或全部结果。结果验证服务不提供对差错响应的验证。

当一个 DSA 处理任意个 DSA（每一个 DSA 同某一个特定服务请求有关联）的结果时，在该 DSA 的上下文内可能有两种不同情况：

— DSA 提供全部操作结果，而不需要进行任何逻辑相加（在图 C-1/X.518 中以 DSA“d”和 DSA“e”表示）；

— DSA 把本地结果（来源于此 DSA 本身）同其他 DSA 送来的结果合并起来（逻辑相加）（在图 C-1/X.518 中以 DSA“c”表示）；

— DSA 把来自某一个 DSA 的结果，采用连锁方式转移到另一个 DSA 或 DUA；这个过程对结果集毫无影响（在图 C-1/X.518 中以 DSA“b”表示）。

C.7.1 *DSA 结果—无逻辑相加*

本节说明 DSA 在做为唯一向特定操作请求提供结果的源时所扮演的角色，即 DSA 不执行逻辑相加功能。本节同时考虑 DSP 和 DAP 的情况。

C.7.1.1 *DSP*

DSA 可选择执行下列规程中之一：

— 回送不署名的结果，这可表示如下：
  ChR，OR＝连锁操作结果（不署名）
  其中，
  ChR＝连锁结果
  OR＝操作结果；

— 只签署操作结果，这可表示如下；
  ChR，d(OR)＝由 DSA“d”签署的操作结果；

— 只签署连锁操作结果，这可表示如下；
  d(ChR，OR)＝由 DSA“d”签署的连锁操作结果；

— 签署操作结果和连锁操作结果，这可表示如下：
  d{ChR，D{OR}}＝由 DSA“d”签署的操作结果和连锁操作结果。

注 — 在操作结果已被署名的情况下，署名的结果将被送回发起者；在连锁操作结果已被署名的情况下，接收者 DSA 将放弃该签名，使得在把连锁操作结果传递之前能够修改连锁结果自变量。

C.7.1.2 *DAP*

详细情况已在建议 X.511 中说明，这里只提供其摘要以资参照。

DSA 可选择回送不署名的结果，并表示如下：

OR＝操作结果

或回送署名结果如下：

d{OR}＝由 DSA“d”签署的操作结果。

C.7.2 *DSA 结果—包括逻辑相加*

本节说明 DSA 在回送特定服务请求的结果时所起的作用。该结果以合并和综合从其他 DSA 获得的结果为必要前提。本节考虑 DSP 和 DAP 两种情况。

C.7.2.1 *DSP*

考虑到可以签署来自其他 DSA 的零个或多个结果，本规程允许 DSA 对结果进行合并和综合、签署综合结果的零个或多个组成部分，并可任意选择签署整体综合结果。

C.7.2.1.1 *连锁结果自变量的生成*

本规程要求 DSA（在图 C-1/X.518 中以 DSA“c”表示）把全部连锁操作结果的签名从由外部 DSA（在图 C-1/X.518 中以 DSA“d”和 DSA“e”表示）收到的结果中删除。这样，DSA“c”拥有一组不署名的连锁结果、一组署名的操作结果和一组不署名的操作结果。

全部连锁结果遵照本建议 §12.4 的描述进行处理，建立一个经修改的连锁结果如下：

i)ChR′＝修改过的连锁结果。

C.7.2.1.2 *不署名的本地结果*

倘若 DSA 不希望签署在本地产生的结果，则把一组不署名的操作结果同本地结果合并，以组成一组修改过的操作结果如下：

OR′＝合并的操作结果。

一组完整的操作结果是由一组外部署名的操作结果组成的逻辑“或”：

d{OR}，e{OR}…

以及合并的操作结果，集体以下式表示：

ii)OR′，d{OR}，e{OR}，…＝操作结果。

C.7.2.1.3 *署名的本地结果*

倘若 DSA 希望签署本地产生的结果，则首先应把外部产生的一组不署名的操作结果合并在一起。一组完整的操作结果是由一组本地签署的操作结果 C{OR}、一组合并的外部不署名的操作结果 OR”以及一组外部签署的操作结果组成的逻辑“或”，表示如下：

d{OR}，e{OR}，…，集体用下列符号表示：

iii)c{OR}，OR″，d{OR}，e{OR}，…＝操作结果。

C.7.2.1.4 *不署名的连锁操作结果*

倘若 DSA 不希望签署连锁操作结果，则后者将包括连锁结果（见上面的(i)）加上在(ii)或(iii)指出的操作结果，集体表示如下：

ChR′,OR′,d{OR},e{OR},…＝连锁操作结果（不署名）。

或，

ChR′,c{OR},OR″,d{OR},e{OR},…＝连锁操作结果（不署名）和由 DSA“c”签署的操作结果。

C.7.2.1.5 *署名的连锁操作结果*

倘若 DSA 希望签署连锁操作结果，则该结果将包括连锁结果（见上面的(i)）加上操作结果（见上面的(ii)或(iii)），集体表示如下：

c{ChR′,OR′,d{OR},e{OR},…}＝由 DSA“c”签署的连锁操作结果

或，

c{ChR′,c{OR},OR″,d{OR},e{OR},…}＝由 DSA“c”签署的连锁操作结果和操作结果。

C.7.2.2 *DAP*

本规程同§C.7.2.1 所述的规程非常相似，唯一的例外是连锁结果自变量没有在 DAP 中传递。

C.7.3 *DSA 连锁结果*

本节说明 DSA 分别在 DSP 和 DAP 内，把操作结果用连锁方式送回请求者 DSA 或 DUA 时使用的规程。

C.7.3.1 *DSP*

DSA 首先从连锁操作结果中删除签名（若存在的话）。然后按照本建议所述对连锁结果自变量进行操作，生成一个修改的连锁结果自变量，然后把连锁结果自变量同操作结果自变量合并，生成一个修改的连锁操作结果。最后，在把连锁操作结果传给链中的下一个 DSA 之前，DSA 可以任意选择签署该结果。

C.7.3.2 *DAP*

DSA（在图 C-1/X.518 中以 DSA“b”表示）首先把签名（若存在一个的话）从连锁操作结果中删除。然后分析和放弃连锁结果自变量。最后，DSA 在把结果传给 DUA 之前任意选择签署其余的操作结果自变量。

附 件 D

（附于建议 X.518）

分布式号码簿客体标识符

本附件是本建议的一部分。

本附件包括在本建议中出现的所有 ASN.1 客体标识符，并以 ASN.1 模块 **Distributed-DirectoryObjectIdentifiers** 的形式提供。

```
DistributedDirectoryObjectIdentifiers        {joint-iso-ccitt ds(5) modules(1)
                                              distributedDirectoryObjectIdentifiers(13)}

DEFINITION::=
BEGIN

EXPORTS
      id-ot-dsa, id-pt-chainedRead, id-pt-chainedSearch, id-pt-chainedModify;

IMPORTS
      id-ot, id-pt
            FROM     UsefulDefinitions {joint-iso-ccitt ds(5) modules(1) usefulDefinitions(0)};

-- objects --

id-ot-dsa    OBJECT IDENTIFIER  ::=  {id-ot 3}

-- part types --

id-pt-chainedRead     OBJECT IDENTIFIER     ::=    {id-pt 4}
id-pt-chainedSearch   OBJECT IDENTIFIER     ::=    {id-pt 5}
id-pt-chainedModify   OBJECT IDENTIFIER     ::=    {id-pt 6}

END
```

建　议　X.519

号码簿协议描述[1]

（1988 年订于墨尔本）

目　录

0　绪论 …… 175

1　范围 …… 175

2　参考文献 …… 176

3　定义 …… 176

3.1　OSI 参考模型定义 …… 176
3.2　基本号码簿定义 …… 176
3.3　分布式操作定义 …… 176

4　缩写词 …… 177

5　规约 …… 177

① 建议 X.519 和 ISO9594－5 号码簿协议描述是在紧密合作下制定的，而且在技术上是协调的。

6 协议概述 …… 177

6.1 号码簿协议模型 …… 177
6.2 号码簿访问协议 …… 178
6.3 号码簿系统协议 …… 178
6.4 底层服务的使用 …… 178

7 号码簿协议抽象语法 …… 179

7.1 抽象语法 …… 179
7.2 号码簿应用服务要素 …… 179
7.3 号码簿应用上下文 …… 181
7.4 差错 …… 182

8 在使用的服务上的映射 …… 182

8.1 在 ACSE 上的映射 …… 182
8.2 在 ROSE 上的映射 …… 183

9 一致性 …… 183

9.1 DUA 的一致性 …… 183
9.2 DSA 的一致性 …… 184

附件 *A* — 使用 ASN.1 描述 DAP …… 185
附件 *B* — 使用 ASN.1 描述 DSP …… 187
附件 *C* — 协议客体标识符的参考定义 …… 188

0 绪论

0.1 本文和本系列其他文件的目的在于简化信息处理系统之间的互连，以便提供号码簿服务。全套系统，包括它们的号码簿信息，可视为一个整体，叫做号码簿。号码簿包含的全部信息叫做号码簿信息库(DIB)，其典型用途是简化客体之间以及同这些客体、或围绕这些客体(如应用实体、人员、终端和分布表)的通信联系。

0.2 号码簿在开放系统互连中扮演重要角色；开放系统的目的，是除使用互连标准之外设法只通过很少的技术协议就能实现下列信息处理系统之间的互连：

— 不同厂家；
— 不同管理部门；
— 不同复杂程度；以及
— 不同年代。

0.3 本建议为两种协议—号码簿访问协议(DAP)和号码簿系统协议(DSP)—说明应用服务要素和应用上下文。DAP 提供访问号码簿、检索或修改号码簿信息的协议。DSP 提供把检索或修改号码簿信息的请求连锁转移到分布式号码簿系统的其他部分 — 存放号码簿信息的地方。

1 范围

本建议描述号码簿访问协议和号码簿系统协议；这些协议的目的是实现建议 X.511 和 X.518 描述的抽象服务。

2 参考文献

建议 X.200 — 开放系统互连基本参考模型
建议 X.208 — 开放系统互连抽象语法符号第 1 集(ASN.1)的规范
建议 X.209 — 开放系统互连抽象语法符号第 1 集(ASN.1)的基本编码规则的规范
建议 X.500 — 号码簿概念、模型和服务概述
建议 X.501 — 号码簿模型
建议 X.511 — 号码簿抽象服务定义
建议 X.518 — 号码簿分布式操作规程
建议 X.520 — 号码簿属性类型集
建议 X.521 — 号码簿客体类别集
建议 X.219 — 远程操作模型、符号和服务定义
建议 X.229 — 远程操作协议描述
建议 X.217 — 开放系统互连的联系控制:服务定义
建议 X.227 — 开放系统互连的联系控制:协议描述
建议 X.216 — 开放系统互连的表示层服务定义

3 定义

本节中的定义使用 § 4 定义的缩写词。

3.1 *OSI* 参考模型定义

本建议是根据建议 X.200 描述的概念制定的,并使用该建议定义的下列术语:

a) *application—service—element* 应用服务要素;
b) *application—protocol—control—information* 应用协议控制信息;
c) *application—control—clata—unit* 应用控制数据单元;
d) *application—context* 应用上下文;
e) *application—entity* 应用实体;
f) *abstract—syntax* 抽象语法。

3.2 基本号码簿定义

本建议使用建议 X.501 定义的下列术语:

a) *the Directory* 号码簿;
b) (*Directory*)*user*(号码簿)用户;
c) *Directory System Agent*(*DSA*) 号码簿系统代理(*DSA*);
d) *Directory User Agent*(*DUA*) 号码簿用户代理(*DUA*)。

3.3 分布式操作定义

本建议使用建议 X.518 定义的下列术语:

a) *chaining* 连锁转移;
b) *referral* 转移参考。

4 缩写词

本建议使用下列缩写词：

AC	应用上下文
ACSE	关联控制服务要素
AE	应用实体
APCI	应用协议控制信息
APDU	应用协议数据单元
ASE	应用服务要素
DAP	号码簿访问协议
DSA	号码簿系统代理
DSP	号码簿系统协议
DUA	号码簿用户代理
ROSE	远程操作服务要素

5 规约

本建议使用下列规约：

a) §7 中的抽象语法定义是使用建议 X.208 定义的抽象语法符号规则定义的；

b) 远程操作宏(RO－符号法)、应用服务要素和应用上下文宏都在建议 X.219 中定义；

c) 定义的术语词、服务参数和协议字段的名称和值，除专有名词外，一律用小写字母开头并使用连字符连接，如：defined－term。专有名词以大写字母开头，而且不用连字符连接，如：Proper Name。

6 协议概述

6.1 号码簿协议模型

建议 X.511 定义 DUA 与号码簿之间的抽象服务，支持用户访问号码簿服务。号码簿模型是由 DSA 组成的；DSA 支持特定访问点。建议 X.518 定义在号码簿内部一对 DSA 之间的对话，以支持用户的连锁请求。这些概念用图 1/X.519 说明。

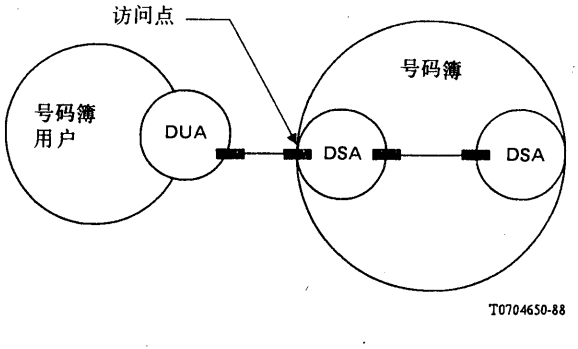


图 1/X.519

号 码 簿 对 话

当参予交互操作的 DUA 和 DSA 不在同一个开放系统时，这些操作是由号码簿访问协议(DAP)支持的，后者是一个 OSI 应用层协议。同理，当进行操作的两个 DSA 不在同一个开放系统时，这样的操作是由号码簿系统协议(DSP)支持的，后者也在应用层。

DAP 和 DSP 是为一对应用进程进行通信提供的协议。在 OSI 环境内，这是由一对应用实体(AE)之间，使用表示服务的通信表示的。AE 的功能是由一组应用服务要素(ASE)提供的。AE 之间的操作，是通过它们使用 ASE 提供的服务说明的。对两个号码簿协议都通用的两个 ASE 将在本节简单介绍。

远程操作服务要素(ROSE)支持在抽象模型中的端口上出现的抽象操作请求/应答范例。号码簿 ASE 提供映射功能：把号码簿抽象服务的抽象语法符号映射在 ROSE 提供的服务上。

关联控制服务要素(ACSE)支持一对 AE 之间的应用关联的建立和释放。DUA 与 DSA 之间的关联只能由 DUA 建立。唯有建立关联的发起者才能释放它。

6.2 号码簿访问协议

号码簿访问协议(DAP)是用于号码簿抽象服务的实施。它包括号码簿的三个特定 ASE，和 ROSE 与 ACSE，即：**readASE**，**searchASE** 和 **modifyASE**。它们分别同抽象服务的 **readPort**，**searchPort** 和 **modifyPort** 相对应。**directoryAccessAC** 应用上下文识别 **readASE**，**searchASE** 和 **modifyASE**，**aCSE**，**rOSE** 的组合。

6.3 号码簿系统协议

号码簿系统协议(DSP)用于实施建议 X.518 描述的号码簿分布式操作功能。它包括号码簿的三个特定 ASE(除 ROSE 和 ACSE 外)，即：**chained chainedReodASE**，**chainedSearchASE** 和 **chainedModifyASE**。它们分别同抽象服务中的 **chainedReadPort**，**chained SearchPort** 和 **chainedModifyPort** 相对应。**directorySystemAC** 应用上下文识别 **chainedReadASE**，**chainedSearchASE** 和 **chainedModifyASE** 的组合。

6.4 底层服务的使用

DAP 和 DSP 协议使用下述底层服务。

6.4.1 ROSE 服务的使用

远程操作服务要素(ROSE)在建议 X.219 中定义。

ROSE 支持远程操作的请求/应答范例。

号码簿 ASE 是下列 ROSE 服务的用户：RO-INVOKE，RO-RESULT，RO-ERROR，RO-REJECT-U 和 RO-REJECT-P 服务。

DAP 和 DSP 的远程操作属于第 2 类(异步)操作。注意，由于 DUA 是 DAP 的用户，它可以选择同步操作方式。

DAP 使用第 1 类关联。这意味着，DSA 不能起动 DUA 的操作。DSP 使用第 3 类关联。这意味着，响应者 DSA 能够起动发起者 DSA 的操作，反之亦然。

6.4.2 *ACSE* 服务的使用

关联控制服务要素(ACSE)在建议 X.217 中定义。

ACSE 提供对 AE 之间的应用关联(application-association)的控制(建立、释放、放弃)。

号码簿 Bind 和号码簿 Unbind(或 DSA Bind 和 DSA Unbind)是下列 ACSE 服务的唯一用户，即：A-ASSOCIATE 和 A-RELEASE 正常方式下的服务。应用一进程是下列 ACSE 服务的用户：A-ABORT 和 A-P-ABORT 服务。

6.4.3 表示服务的使用

表示服务在建议 X.216 中定义。

表示层协调拟交换的应用层语义学的代表(语法)。

在正常方式下,对包含在应用上下文内的每一个抽象语法应分别使用不同的表示上下文。

ACSE 是下列表示服务的唯一用户:P-CONNECT,P-RELEASE,P-U-ABORT 和 P-P-ABORT 服务。

ROSE 是表示服务的 P-DATA 服务的用户。

6.4.4 低层服务的使用

会话层服务在建议 X.215 中定义。会话层建立最终系统之间的信息流对话。

会话服务的核心(Kernel)与双工(Duplex)功能单元为表示层提供服务。

传送服务在建议 X.214 中定义。传送层在低层的网络连接上提供端至端透明数据传送服务。

会话层使用的传送服务级别,是根据对多路复用和差错恢复的要求选择的。对传送级别 0(不复用)的支持是强制性的。在这里不使用传送加急服务。

对其他传送级别的支持是任选的。多路复用级别可用于在同一条网络连接上对 DAP、DSP 和其他协议的复用。差错恢复级别可在剩余误码率高得不能接受的网络连接上选择使用。

支持建议 X.213 定义的 OSI 网络服务使用的低层网络是假设提供的。

网络地址在建议 X.121、E.163/E.164 或 X.200(OSI NSAP-地址)中定义。

7 号码簿协议抽象语法

7.1 抽象语法

在§7.2.1,§7.2.3 和§7.2.5 中说明的号码簿 ASE,共享一个抽象语法:**id-as-directory-AccessAS**。在§7.2.2,§7.2.4 和§7.2.6 中说明的号码簿 ASE 亦共享一个抽象语法:**id-as-directorySystemAS**。在以上两种情况中,抽象语法定义应用协议控制信息(APCI),当后者与 ROSE 结合使用时,定义一组 APDU。号码簿 APDU 是由号码簿 ASE 和 ROSE 的抽象语法定义的。这些加上 ACSE 的抽象语法构成 APDU 的完整定义,在号码簿的关联过程要用到这些 APDU。

ACSE 抽象语法 **id-as-acse** 是在建立关联时需要用的。

上述抽象语法必须(做为最低限度的要求)按照 ASN.1 的基本编译规则进行编译。

7.2 号码簿应用服务要素

本节描述在建立各种号码簿应用上下文(如§7.3 所述)中拿来做"材料"的 ASE。

注 — 这些 ASE 用来建立在本建议中定义的应用上下文。它们的目的不是为了同个别 ASE 或其他 ASE 组合保持一致。

7.2.1 读 *ASE*

readASE 支持 **readPort** 的抽象操作,即建议 X.511 中定义的 **Read**,**Compare** 和 **Abandon**。

```
readASE
    APPLICATION-SERVICE-ELEMENT
        CONSUMER INVOKES
            {read, compare, abandon}
    ::=  id-ase-readASE

read        Read      ::= 1

compare     Compare   ::= 2

abandon     Abandon   ::= 3
```

7.2.2 连锁读 *ASE*

Chained ReadASE 支持 **ChainedReadPort** 的抽象操作，即建议 X.518 所定义的 **ChainedRead**，**ChainedCompare** 和 **ChainedAbandon**。

```
chainedReadASE
    APPLICATION-SERVICE-ELEMENT
        OPERATIONS {
            chainedRead,
            chainedCompare
            chainedAbandon}
    ::=  id-ase-chainedReadASE

chainedRead       ChainedRead      ::= 1

chainedCompare    ChainedCompare   ::= 2

chainedAbandon    ChainedAbandon   ::= 3
```

7.2.3 搜索 *ASE*

searchASE 支持 **SearchPort** 的抽象操作，即建议 X.511 定义的 **List** 和 **Search**.

```
searchASE
    APPLICATION-SERVICE-ELEMENT
        CONSUMER INVOKES { list, search}
    ::=  id-ase-searchASE}

list        List    ::= 4

search      Search  ::= 5
```

7.2.4 连锁搜索 *ASE*

chainedSearchASE 支持 **ChainedSearchPort** 的抽象操作，即建议 X.518 定义的 **ChainedList** 和 **ChainedSearch**。

```
chainedSearchASE
    APPLICATION-SERVICE-ELEMENT
        OPERATIONS {
            chainedList, chainedSearch}
    ::=  id-ase-chainedSearchASE

chainedList       ChainedList      ::= 4

chainedSearch     ChainedSearch    ::= 5
```

7.2.5 修改 *ASE*

modifyASE 支持 **ModifyPort** 的抽象操作，即建议 X.511 定义的 **AddEntry**，**RemoveEntry**，**ModifyEntry** 和 **ModifyRDN**。

```
modifyASE
    APPLICATION-SERVICE-ELEMENT
        CONSUMER INVOKES
            {addEntry, removeEntry,
            modifyEntry, modifyRDN}
    ::=  id-ase-modifyASE
addEntry        AddEntry        ::= 6
removeEntry     RemoveEntry     ::= 7
modifyEntry     ModifyEntry     ::= 8
modifyRDN       ModifyRDN       ::= 9
```

7.2.6 连锁修改 *ASE*

chainedModifyASE 支持 **ChainedModifyPort** 的抽象操作，即建议 X.518 定义的 **ChainedAddEntry**，**ChainedRemoveEntry**，**ChainedModifyEntry** 和 **ChainedModifyRDN**。

```
chainedModifyASE
    APPLICATION-SERVICE-ELEMENT
        OPERATIONS
            {chainedAddEntry,
             chainedRemoveEntry,
             chainedModifyEntry,
             chainedModifyRDN}
    ::=  id-ase-chainedModifyASE
chainedAddEntry         ChainedAddEntry         ::= 6
chainedRemoveEntry      ChainedRemoveEntry      ::= 7
chainedModifyEntry      ChainedModifyEntry      ::= 8
chainedModifyRDN        ChainedModifyRDN        ::= 9
```

7.3 号码簿应用上下文

7.3.1 号码簿访问应用上下文

directoryAccessAC 允许 DUA 访问下列 ASE 的操作：**readASE**，**searchASE**，**modifyASE**。

```
directoryAccessAC
    APPLICATION-CONTEXT
        APPLICATION SERVICE ELEMENTS
            {aCSE}
        BIND DirectoryBind
        UNBIND DirectoryUnbind
        REMOTE OPERATIONS {rOSE}
        INITIATOR CONSUMER OF {
            readASE,
            searchASE,
            modifyASE}
        ABSTRACT SYNTAXES {
            id-as-acse,
            id-as-directoryAccessAS}
    ::=  id-ac-directoryAccessAC
```

7.3.2 号码簿系统应用上下文

directorySystemAC 允许各 DSA 为连锁操作目的进行通信。

```
directorySystemAC
    APPLICATION-CONTEXT
        APPLICATION SERVICE ELEMENTS
            {aCSE}
        BIND DSABind
        UNBIND DSAUnbind
```

```
        REMOTE OPERATIONS {rOSE}
        OPERATIONS OF
            {chainedReadASE,
            chainedSearchASE,
            chainedModifyASE}
        ABSTRACT SYNTAXES {
            id-as-acse,
            id-as-directorySystemAS}
::=     id-ac-directorySystemAC
```

7.4 差错

对应每一个在抽象服务中定义的抽象差错就有一个差错值，可由协议进行传递。其赋值如下：

abandoned	**Abandoned**	**::= 5**
attributeError	**AttributeError**	**::= 1**
nameError	**NameError**	**::= 2**
referral	**Referral**	**::= 4**
securityError	**SecurityError**	**::= 6**
serviceError	**ServiceError**	**::= 3**
updateError	**UpdateError**	**::= 8**
dSAReferral	**DSAReferral**	**::= 9**
abandonFailed	**AbandonFailed**	**::= 7**

8 在使用的服务上的映射

本节定义 DAP 和 DSP 在使用的服务上的映射。

8.1 在 *ACSE* 上的映射

本节定义抽象耦合(**DirectoryBind** 或 **DSABind**)和抽象释放(**DirectoryUnbind** 或 **DSAUnbind**)服务在 ACSE 服务上的映射。ACSE 在建议 X.217 中定义。

8.1.1 抽象耦合在 *A-ASSOCIATE* 上的映射

抽象耦合服务是在 ACSE 的 A-ASSOCIATE 服务上映射的。A-ASSOCIATE 服务参数的使用将在下面的分章节里描述。

8.1.1.1 方式

此参数应由关联的发起者在 A-ASSOCIATE 请求原语中提供，并应取“正常方式”值。

8.1.1.2 应用上下文名

关联的发起者应提议采用 **directoryAccessAC** 或 **directorySystemAC** 应用上下文。

8.1.1.3 用户信息

抽象耦合服务的耦合操作在 A-ASSOCIATE 请求原语的用户信息参数上的映射在建议 X.219 中定义。

8.1.1.4 表示上下文定义表

关联的发起者应在 A-ASSOCIATE 请求原语中提供表示上下文定义表，表中应包括 ACSE 抽象语法(**id-as-acse**)和 DAP 抽象语法(**id-as-directoryAccessAS**)或 DSP 抽象语法(**id-as-directorySystemAS**)。

8.1.1.5 服务质量

此参数应由关联的发起者在 A-ASSOCIATE 请求原语中提供，和由关联的响应者在 A-ASSOCIATE 响应原语中提供。参数"扩展的控制"(Extended Control)和"优化的对话传递"(Optimized Dialogue Transfer)应置为"此功能不需要"。其余的参数则应使用默认值。

8.1.1.6 会话要求

此参数应由关联的发起者在 A-ASSOCIATE 请求原语中设置，和由关联的响应者在 A-ASSOCIATE 响应原语中设置。此参数应置为下列功能单元：

a) 核心；

b) 双工。

8.1.1.7 应用实体名称和表示地址

这些参数应由关联的发起者和响应者提供(应用实体名称是任选的)。为第一次请求建立关联的 DUA，可从本地保存的信息中获得这些参数。

对于与参考中指定的 DSA 建立关联的 DUA(或 DSA)而言，这些参数可从 **ContinuanceReference** 的值 **AccessPoint** 中获得。建立关联的 DSA 可从它的知识信息，即外部参考中获得这个参数。

8.1.2 抽象释放在 *A-RELEASE* 上的映射

抽象释放服务在 ACSE 的 A-RELEASE 服务上映射。关于 A-RELEASE 服务参数的使用，在下面的分章节中描述。

8.1.2.1 结果

此参数的值应为"肯定"

8.1.3 *A-ABORT* 和 *A-P-ABORT* 服务的使用

应用进程是 ACSE 的服务 A-ABORT 和 A-P-ABORT 的用户。

8.2 在 *ROSE* 上的映射

号码簿 ASE 服务是在 ROSE 服务：RO-INVOKE、RO-RESULT、RO-ERROR、RO-REJECT-U 和 RO-REJECT-P 上映射的。号码簿 ASE 的抽象语法符号在 ROSE 服务上的映射是在建议 X.219 中定义的。

9 一致性

本节定义声明同本建议保持一致的一致性要求。

9.1 *DUA* 的一致性

一个声明同本建议保持一致的 DUA 实现，应满足 § 9.1.1 至 § 9.1.3 提出的各项要求。

9.1.1 对声明的要求

声明中必须包括下列各项：

a)按照一致性声明，指出 DUA 能够起动的 **directoryAccessAC** 应用上下文的操作；
b)符合一致性声明的安全保密级别(无、简单、强)。

9.1.2 静态要求

DUA 应当：
a)具备支持在 § 7 中由其抽象语法定义的 **directoryAccessAC** 应用上下文的能力。

9.1.3 动态要求

DUA 应当：
a)符合 § 8 定义的“在使用的服务上的映射”规程。

9.2 *DSA* 的一致性

一个声明同本建议保持一致的 DSA 实现，应满足 § 9.2.1 至 § 9.2.3 提出的各项要求。

9.2.1 对声明的要求

声明中必须包括下列各项：

a) 符合一致性声明的应用上下文：**directoryAccessAC**，**directorySystemAC**，或二者兼有。假设有一个 DSA，已知有关它的知识已被传播出去，导致其他 DMD 内的其他 DSA 也保存了有关该 DSA 的知识参考，则该 DSA 应声明同 **directorySystemAC** 的一致性。
注 — 除在此申明者外，应用上下文是不应当加以分割的。特别是不能针对某一些端口或操作申明一致性。
b) 申明 DSA 是否能够担任建议 X.518 定义的一级 DSA；
c) 假设已经申明对 **directorySystemAC** 应用上下文的一致性，则应说明是否支持建议 X.518 定义的连锁操作方式；
d) 符合一致性声明的安全保密级别(无、简单、强)；
e) 建议 X.520 定义的属性类型集以及符合一致性申明的其他属性类型；
f) 建议 X.521 定义的客体类别集以及符合一致性声明的其他客体类别。

9.2.2 静态要求

DSA 应当：

a) 具备支持符合一致性申明和由 § 7 的抽象语法定义的应用上下文的能力；
b) 具备支持由建议 X.501 的抽象语法定义的信息框架的能力；
c) 符合建议 X.518 定义的最低限度知识的要求；
d) 假设申明对一级 DSA 的一致性，则应符合对建议 X.518 定义的根上下文的支持的要求；
e) 具备支持使用抽象语法定义、符合一致性申明的属性类型的能力；和
f) 具备支持使用抽象语法定义、符合一致性申明的客体类别的能力。

9.2.3 动态要求

DSA 应当：

a) 符合本建议 § 8 定义的“在使用的服务上的映射”规程；

b) 符合建议 X.518 定义的转移参考号码簿分布式操作规程；

c) 假设申明对 **directoryAccessAC** 应用上下文的一致性，则应符合建议 X.518 有关使用转移参考方式的 DAP 规程；

d) 假设申明对 **directorySystemAC** 应用上下文的一致性，则应符合建议 X.518 定义的转移参考交互式操作规程；

e) 假设申明对连锁交互式操作的一致性，则应符合建议 X.518 定义的连锁交互式操作规程。

注 — 唯在此情况下，DSA 才需要具备使用 **directorySystemAC** 来起动操作的能力。

附 件 A

（附于建议 X.519）

使用 ASN.1 描述 DAP

本附件是本建议的一部分。

本附件包括本建议中出现的所有 ASN.1 类型和值定义，并以 ASN.1 模块 **DirectoryAccessProtocol** 的形式提供。

```
DirectoryAccessProtocol {joint-iso-ccitt ds(5) modules(1) dap(11)}
DEFINITIONS ::=
BEGIN

EXPORTS
    directoryAccessAC, readASE, searchASE, modifyASE;

IMPORTS
    abstractService
    FROM    UsefulDefinitions
            {joint-iso-ccitt ds(5) modules(1) usefulDefinitions(0)}

    APPLICATION-SERVICE-ELEMENT, APPLICATION-CONTEXT, aCSE
    FROM    Remote-Operations-Notation-extension
            {joint-iso-ccitt remoteOperations(4) notation-extension(2)}

    id-ac-directoryAccessAC, id-ase-readASE, id-ase-searchASE,
    id-ase-modifyASE, id-as-directoryAccessAS, id-as-acse
    FROM    ProtocolObjectIdentifiers
            {joint-iso-ccitt  ds(5) modules(1)
            protocolObjectIdentifiers(4)}

    DirectoryBind, DirectoryUnbind, Read, Compare, Abandon, List,
    Search, AddEntry, RemoveEntry, ModifyEntry, ModifyRDN, Abandoned, AbandonFailed,
    AttributeError, NameError, Referral, SecurityError, ServiceError,
    UpdateError
    FROM    DirectoryAbstractService
            directoryAbstractService;

-- Application Contexts --

directoryAccessAC
    APPLICATION-CONTEXT
        APPLICATION SERVICE ELEMENTS {aCSE}
        BIND DirectoryBind
```

```
    UNBIND DirectoryUnbind
        REMOTE OPERATIONS {rOSE}
            INITIATOR CONSUMER OF {readASE, searchASE, modifyASE}
        ABSTRACT SYNTAXES {
            id-as-acse, id-as-directoryAccessAS}
    ::=  id-ac-directoryAccessAC

-- Read ASE --

readASE
    APPLICATION-SERVICE-ELEMENT
        CONSUMER INVOKES {read, compare, abandon}
    ::=  id-ase-readASE

-- Search ASE --

searchASE
    APPLICATION-SERVICE-ELEMENT
        CONSUMER INVOKES {list, search}
    ::=  id-ase-searchASE

-- Modify ASE --

modifyASE
    APPLICATION-SERVICE-ELEMENT
        CONSUMER INVOKES
            {addEntry, removeEntry,
             modifyEntry, modifyRDN}
    ::=  id-ase-modifyASE

-- Remote Operations --

read            Read            ::= 1
compare         Compare         ::= 2
abandon         Abandon         ::= 3
list            List            ::= 4
search          Search          ::= 5
addEntry        AddEntry        ::= 6
removeEntry     RemoveEntry     ::= 7
modifyEntry     ModifyEntry     ::= 8
modifyRDN       ModifyRDN       ::= 9

-- Remote Errors --

attributeError  AttributeError  ::= 1
nameError       NameError       ::= 2
serviceError    ServiceError    ::= 3
referral        Referral        ::= 4
abandoned       Abandoned       ::= 5
securityError   SecurityError   ::= 6
abandonFailed   AbandonFailed   ::= 7
updateError     UpdateError     ::= 8
END
```

附 件 B

（附于建议 X. 519）

使用 ASN. 1 描述 DSP

本附件是本建议的一部分。

本附件包括本建议出现的所有 ASN. 1 类型和值定义，并以 ASN. 1 模块 **DirectorySystemProtocol** 的形式提供。

```
DirectorySystemProtocol {joint-iso-ccitt ds(5) modules(1) dsp(12)}
DEFINITIONS ::=
BEGIN

EXPORTS
    directorySystemAC, chainedReadASE, chainedSearchASE, chainedModifyASE;

IMPORTS
    distributedOperations, directoryAbstractService
    FROM    UsefulDefinitions
            {joint-iso-ccitt ds(5) modules(1) usefulDefinitions(0)}

    APPLICATION-SERVICE-ELEMENT, APPLICATION-CONTEXT, aCSE
    FROM    Remote-Operations-Notation-extension
            {joint-iso-ccitt remoteOperations(4) notation-extension(2)}

    id-ac-directorySystemAC, id-ase-chainedReadASE,
    id-ase-chainedSearchASE, id-ase-chainedModifyASE,
    id-as-directorySystemAS, id-as-acse;
    FROM    ProtocolObjectIdentifiers
            {joint-iso-ccitt  ds(5) modules(1)
            protocolObjectIdentifiers(4)}

    Abandoned, AttributeError, AbandonFailed,
    NameError, DSAReferral, SecurityError, ServiceError, UpdateError
    FROM    DirectoryAbstractService directoryAbstractService

    DSABind, DSAUnbind,
    ChainedRead, ChainedCompare, ChainedAbandon,
    ChainedList, ChainedSearch,
    ChainedAddEntry, ChainedRemoveEntry, ChainedModifyEntry,
    ChainedModifyRDN, DSAReferral,
    FROM    DistributedOperations
            distributedOperations;

-- Application Contexts --

directorySystemAC
    APPLICATION-CONTEXT
        APPLICATION SERVICE ELEMENTS {aCSE}
        BIND DSABind
        UNBIND DSAUnbind
        REMOTE OPERATIONS {rOSE}
            OPERATIONS OF {
                chainedReadASE, chainedSearchASE, chainedModifyASE}
        ABSTRACT SYNTAXES {
            id-as-acse, id-as-directorySystemAS}
    ::=  {id-ac-directorySystemAC}

-- Chained Read ASE --

chainedReadASE
    APPLICATION-SERVICE-ELEMENT
        OPERATIONS      {chainedRead, chainedCompare, chainedAbandon}
::=  id-ase-chainedReadASE
```

```
-- Chained Search ASE --

chainedSearchASE
    APPLICATION-SERVICE-ELEMENT
        OPERATIONS {chainedList, chainedSearch}
    ::=  id-ase-chainedSearchASE

-- Chained Modify ASE --

chainedModifyASE
    APPLICATION-SERVICE-ELEMENT
        OPERATIONS
            {chainedAddEntry, chainedRemoveEntry,
            chainedModifyEntry, chainedModifyRDN}
    ::=  id-ase-chainedModifyASE

-- Remote Operations --

chainedRead             ChainedRead             ::= 1
chainedCompare          ChainedCompare          ::= 2
chainedAbandon          ChainedAbandon          ::= 3
chainedlist             ChainedList             ::= 4
chainedSearch           ChainedSearch           ::= 5
chainedAddEntry         ChainedAddEntry         ::= 6
chainedRemoveEntry      ChainedRemoveEntry      ::= 7
chainedModifyEntry      ChainedModifyEntry      ::= 8
chainedModifyRDN        ChainedModifyRDN        ::= 9

-- Remote Errors --

attributeError      AttributeError      ::= 1
nameError           NameError           ::= 2
serviceError        ServiceError        ::= 3
abandoned           Abandoned           ::= 5
securityError       SecurityError       ::= 6
abandonFailed       AbandonFailed       ::= 7
updateError         UpdateError         ::= 8
dsaReferral         DSAReferral         ::= 9
END
```

附 件 C

（附于建议 X. 519）

协议客体标识符的参考定义

本附件是本建议的一部分。

本附件包括本建议中分配的所有 ASN. 1 客体标识符，并以 ASN. 1 模块 **ProtocolObjectIdentifiers** 的形式提供。

```
ProtocolObjectIdentifiers {joint-iso-ccitt ds(5) modules(1)  protocolObjectIdentifiers(4)}
DEFINITIONS ::=
BEGIN
```

```
EXPORTS
    id-ac-directoryAccessAC, id-ac-directorySystemAC, id-ase-readASE, id-ase-searchASE,
    id-ase-modifyASE, id-ase-chainedReadASE,
    id-ase-chainedSearchASE, id-ase-chainedModifyASE, id-as-acse,
    id-as-directoryAccessAS, id-as-directorySystemsAS;

IMPORTS
    id-ac, id-ase, id-as
    FROM    UsefulDefinitions
            {joint-iso-ccitt ds(5) modules(1) usefulDefinitions(0)};

-- Application Contexts --

id-ac-directoryAccessAC   OBJECT IDENTIFIER ::= {id-ac 1}

id-ac-directorySystemAC   OBJECT IDENTIFIER ::= {id-ac 2}

-- ASEs --

id-ase-readASE            OBJECT IDENTIFIER ::= {id-ase 1}

id-ase-searchASE          OBJECT IDENTIFIER ::= {id-ase 2}

id-ase-modifyASE          OBJECT IDENTIFIER ::= {id-ase 3}

id-ase-chainedReadASE     OBJECT IDENTIFIER ::= {id-ase 4}

id-ase-chainedSearchASE   OBJECT IDENTIFIER ::= {id-ase 5}

id-ase-chainedModifyASE   OBJECT IDENTIFIER ::= {id-ase 6}

-- ASs --

id-as-directoryAccessAS   OBJECT IDENTIFIER ::= {id-as 1}

id-as-directorySystemAS   OBJECT IDENTIFIER ::= {id-as 2}

id-as-acse                OBJECT IDENTIFIER ::=
                          {joint-iso-ccitt association-control(2) abstract-syntax(1) apdus(0)
                          version1(1)}

END
```

建　议　X. 520

号码簿属性类型集[①]

（1988 年订于墨尔本）

目　录

0　概论 …… 190

1　应用范围 …… 191

2　参考文献 …… 191

3　定义 …… 191

4　符号法 …… 191

① 建议 X. 520 和 ISO9594-6 信息处理系统—开放系统互连—号码簿属性类型集是在紧密合作下制定的，而且在技术上是协调的。

第 1 部分 — 属性类型集 …… 191

5 属性类型集的定义 …… 191

5.1 系统属性类型 …… 191
5.2 标签属性类型 …… 192
5.3 地理属性类型 …… 193
5.4 组织属性类型 …… 194
5.5 说明性属性类型 …… 195
5.6 邮政寻址属性类型 …… 196
5.7 电信寻址属性类型 …… 197
5.8 优先选择属性类型 …… 199
5.9 OSI 应用属性类型 …… 200
5.10 关系属性类型 …… 200
5.11 安全保密属性类型 …… 201

第 2 部分 — 属性语法 …… 201

6 属性语法定义 …… 201

6.1 号码簿使用的属性语法 …… 201
6.2 字符串属性语法 …… 202
6.3 其他属性语法 …… 203

附件 *A* — 使用 ASN. 1 描述的属性类型集 …… 204

附件 *B* — 属性类型和语法索引 …… 210

附件 *C* — 上限 …… 211

0 概论

0.1 本文和本系列其它文件的目的在于简化信息处理系统之间的互连，以便提供号码簿服务。全套系统，包括它们的号码簿信息，可视为一个整体，叫做号码簿。号码簿包含的全部信息叫做号码簿信息库(DIB)，其典型用途是简化客体之间以及同这些客体，或围绕这些客体(如应用实体、人员、终端和分布表)的通信联系。

0.2 号码簿在开放系统互连中扮演重要角色；开放系统的目的是除使用互连标准之外，设法只通过很少的技术协议就能实现下列信息处理系统之间的互连：

— 不同厂家；
— 不同管理部门；
— 不同复杂程度；
— 不同年代。

0.3 本建议定义一些在一系列号码簿应用中可能要用到的属性类型。在这里定义的许多属性的一个用途是用于名称的构造，特别是建议 X.521 定义的客体类别的名称。此外，本建议还定义一些标准属性语法。

0.4 附件 A 是本建议的一部分，它提供以 ASN.1 符号法为基础的属性和属性语法的全部定义模块。

0.5 附件 B 不是本建议的一部分，它提供属性类型的字母索引，以便于参考。

1 应用范围

1.1 本建议定义一些在一系列号码簿应用中可能要用到的属性类型。

1.2 属性类型(和属性语法)分为三大类,如§1.2.1至§1.2.3中所述。

1.2.1 有些属性类型(语法)使用于种类繁多的应用中,或为号码簿本身所掌握和/或由它本身使用。

注 — 建议在适当场合使用本建议定义的属性类型(语法),而不要生成新的属性类型和语法。

1.2.2 有些属性类型(语法)已国际标准化,但它们是按应用分类的,并在有关的标准中定义。

1.2.3 任何行政权威都能够为任何目的定义自己的属性类型(语法)。这些类型和语法不属于国际标准,行政权威以外的其他单位只能获得通过双边协议建立的属型类型。

2 参考文献

ISO3166 — 表示国家名称的代码
建议 X.121 — 公用数据网的国际编号方案
建议 X.208 — 开放系统互连抽象语法符号第1集(ASN.1)的规范(参见 ISO8824)
建议 X.501 — 号码簿模型(参见 ISO9594-2)
建议 X.521 — 号码簿客体类别集(参见 ISO9594-7)
建议 E.123 — 国内和国际电话号码表示法

3 定义

本建议使用建议 X.501 定义的下列各项:

a) *attribute type* 属性类型;
b) *attribute syntax* 属性语法;
c) *object class* 客体类别。

4 符号法

本文件使用特殊符号法(即在建议 X.501 中定义的 ASN.1 宏)定义属性类型和属性语法。有两种这样的宏,即 **ATTRIBUTE** 和 **ATTRIBUTE-SYNTAX**。

在分别为属性类型和属性语法定义客体标识符时,使用了两种"通用"客体标识符(**attributeType** 和 **attributeSyntax**)。其定义详见建议 X.501 中的附件 B。

说明使用属性类型的范例,采用的是非规范化的描述方法。其中,属性类型和相关的值是用属性类型的缩写表示的,后随等号("="),并后随属性的范例值。

第1部分 — 属性类型集

5 属性类型集的定义

本建议定义一些在一系列号码簿应用中可能要用到的属性类型。

5.1 系统属性类型

这些属性类型涉及号码簿所掌握的客体的相关情报。

5.1.1 *Object Class* 客体类别

号码簿所掌握的客体类别属性类型，除客体标识符的分配外，已在建议 X.501 中描述。

objectClass ObjectClass ::= {attributeType 0}

5.1.2 *Aliased Object Name* 别名客体名

此属性类型，除客体标识符的分配外，已在建议 X.501 中定义。

aliasedObjectName AliasedObjectName ::= {attributeType 1}

5.1.3 *knowledge information* 知识信息

知识信息属性类型描述特定 DSA 所掌握的，人们所能阅读的知识的累积说明。

```
knowledgeInformation ATTRIBUTE
    WITH ATTRIBUTE-SYNTAX caseIgnoreStringSyntax
    ::= {attributeType 2}
```

5.2 标签属性类型

这些属性类型涉及有关客体的情报，通过标签操作使该客体与上述情报建立显关联。

5.2.1 *Common Name* 普通名称

普通名称属性类型描述一个客体的标识符。普通名称不是号码簿名称；它仅是这样一个名称（可能是很含糊的），通过这个名称可以在有限范围内认识这个客体（例如在一个组织范围内），并符合有关国家的文化和命名规范。

普通名称的属性值是一个字符串，可由它所述的人或组织、或由负责该客体的组织为某些装置和应用实体选择。例如，在讲英语的国家，一个人的典型名称包括头衔（例如，先生、女士、博士、教授、阁下、爵士）、名、中间名、姓，和表示后裔的限定词（若有，例如，Jr.）、荣誉勋章和荣誉奖状（若有，例如 QC）。

例如：

CN = "Mr. Robin Lachlan McLeod BSc(Hons) CEng MIEE"

CN = "Divisional Coordination Committee"

CN = "Higb Speed Modem".

任何派生的普通名称必须做为独立和另选值同被指名客体相关联。

其他派生的普通名称也应当考虑，例如使用中间名代替名；使用“Bill”代替“william”，等等。

```
commonName ATTRIBUTE
    WITH ATTRIBUTE-SYNTAX
        caseIgnoreStringSyntax
            (SIZE(1..ub-common-name))
    ::= {attributeType 3}
```

5.2.2 *Surname* 姓

姓属性类型描述通常由父母继承，或婚姻获得，并皆为人知的语言学结构。

姓的属性值是一个字符串，例如：“McLeod”。

```
surname ATTRIBUTE
    WITH ATTRIBUTE-SYNTAX
        caseIgnoreStringSyntax
            (SIZE(1..ub-surname))
    ::= {attributeType 4}
```

5.2.3 *Serial Number* 系列号

系列号属性类型描述一个标识符，一个装置的系列号。

系列号的属性值是可打印的字符串。

```
serialNumber ATTRIBUTE
    WITH ATTRIBUTE-SYNTAX
        printableStringSyntax
            (SIZE(1..ub-serial-number))
    ::=   (attributeType 5}
```

5.3 地理属性类型

这些属性值涉及与客体有关的地理位置或地区。

5.3.1 *Country Name* 国别名

国别名属性类型表示一个国家。当把它用于号码簿名称，做为其中的一个成分时，国别名指出被指名客体所在的地理位置，或者与该客体有重要关系的国家。

国别名的属性值是从 ISO3166 中挑选的字符串。

```
countryName ATTRIBUTE
    WITH ATTRIBUTE-SYNTAX
        PrintableString (SIZE(2)) - IS 3166 codes only
    MATCHES FOR EQUALITY
    SINGLE VALUE
    ::= {attributeType 6}
```

这种类型使用与 **caseIgnoreStringSyntax** 相同的匹配规则。

5.3.2 *Locallity Name* 地方名

地方名属性类型描述一个地方。当把它用于号码簿名称，做为其中的一个成分时，地方名指出被指名客体所在的地区或地方，或者与该客体有重要关系的地区或地方。

地方名的属性值是一个字符串，例如：L＝“Edinburg”

```
localityName ATTRIBUTE
    WITH ATTRIBUTE-SYNTAX
        caseIgnoreStringSyntax
            (SIZE(1..ub-locality-name))
    ::= {attributeType 7}
```

5.3.3 *State or Province Name* 州或省名

州或省名属性类型描述一个州或省。当把它用于号码簿名称，做为其中的一个成分时，州名或省名指出被指名客体所在的地理划分区，或者与该客体有重要关系的地理划分区。

州或省名的属性值是一个字符串，例如：S＝“Ohio”。

```
stateOrProvinceName ATTRIBUTE
    WITH ATTRIBUTE-SYNTAX
        caseIgnoreStringSyntax
            (SIZE(1..ub-state-name))
    ::= {attributeType 8}
```

5.3.4 *Street Address* 街道地址

街道地址属性类型在邮政地址内指定本地分送和物理投递的地点，即街名、处所、大马路和住宅号码。当它用于号码簿名称，做为其中的一个成分时，它指出被指名客体所在的街道地址，或者与该客体有重要关系的街道。

街道地址的属性值是一个字符串，例如："Arnulfstraβe 60"。

```
streetAddress ATTRIBUTE
    WITH ATTRIBUTE-SYNTAX
        caseIgnoreStringSyntax
            (SIZE(1..ub-street-address))
    ::= {attributeType 9}
```

5.4 组织属性类型

这些属性类型涉及组织，它可通过与该客体有关系的组织来描述该客体。

5.4.1 *Organization Name* 组织名称

组织名称属性类型描述一个组织。当它用于号码簿名称，做为其中的一个成分时，它指出与被指名的客体有关联的组织。

组织名称的属性值是由组织选择的一个字符串(例如：O="Scottish Telecommunications plc")。任何派生的组织名称必须做为独立和另选的属性值，同被指名的组织相关联。

```
organizationName ATTRIBUTE
    WITH ATTRIBUTE-SYNTAX
        caseIgnoreStringSyntax
            (SIZE(1..ub-organization-name))
    ::=  {attributeType 10}
```

5.4.2 *Organization Unit Name* 组织单位名称

组织单位名称属性类型描述一个组织单位。当它用于号码簿名称，做为其中的一个成分时，它指出与被指名的客体有关的组织单位。

指定的组织单位应理解为由组织名称属性指定的一个组织的组成部分。

由此可见，若把组织单位名称属性应用于号码簿名称中，则它必须同一个组织名称属性有关联。

组织单位名称的属性值是由该组织选择的一个字符串(例如：OU="Technology Division")。请注意：常用的缩写词"TD"应当是独立和另选的属性值。

例如：

```
O = "Scottel",OU = "TD"

organizationalUnitName ATTRIBUTE
    WITH ATTRIBUTE-SYNTAX
        caseIgnoreStringSyntax
            (SIZE(1..ub-organizational-unit-name))
    ::=  {attributeType 11}
```

5.4.3 *Title* 头衔

头衔属性类型描述客体在一个组织里指定担任的职位或职务。

头衔的属性值是一个字符串。

例如：

T="Manager,Distributed Applications"

```
title ATTRIBUTE
    WITH ATTRIBUTE-SYNTAX
        caseIgnoreStringSyntax
            (SIZE(1..ub-title))
    ::= {attributeType 12}
```

5.5 说明性属性类型

这些属性类型涉及有关客体的说明(例如:用自然语言)。

5.5.1 *Description* 说明

说明属性类型描述说明有关客体的文件。

例如,客体"Standards Interest"可能有一个这样的说明"distribution list for exchange of information about intra-company standards development"。

说明的属性值是一个字符串。

```
description ATTRIBUTE
    WITH ATTRIBUTE-SYNTAX
        caseIgnoreStringSyntax
            (SIZE(1..ub-description))
    ::= {attributeType 13}
```

5.5.2 *Search Guide* 搜索指南

搜索指南属性类型描述有关推荐的搜索准则的情报,这些情报可列入某些登录项,后者可为搜索操作提供方便的基础客体,例如,国别或组织。

搜索准则包括被寻找的客体类别的任选标识符和将在过滤器中使用的属性类型组合和逻辑算子。对搜索准则的每一项有可能指出匹配的程度,例如:近似匹配。

搜索指南属性可以重复,反映各种类型的请求,例如寻找一个住宅的人员或一个组织内部的人员,这都可以从读该搜索指南的客体,即给定的基础客体中得到满足。

```
searchGuide ATTRIBUTE
    WITH ATTRIBUTE-SYNTAX
        Guide
    ::=  {attributeType 14}

Guide ::= SET {
    objectClass     [0]   OBJECT-CLASS OPTIONAL,
    criteria        [1]   Criteria}

Criteria ::=
    CHOICE{
        Type        [0]   CriteriaItem,
        and         [1]   SET OF Criteria,
        or          [2]   SET OF Criteria,
        not         [3]   Criteria

CriteriaItem
    CHOICE {
        equality            [0] AttributeType,
        substrings          [1] AttributeType,
        greaterOrEqual      [2] AttributeType,
        lessOrEqual         [3] AttributeType,
        approximateMatch    [4] AttributeType}
```

例子:下面是搜索指南属性的一个可能值,这个值可以在客体类别所在地的登录项中储存,指出如何找到客体类别住宅人员的登录项。

```
residential-person-guide  Guide ::= {
      objectClass residentialPerson,
      criteria and {
        type substrings commonName,
        type substrings streetAddress }}
```

使用上述指南的值建立一个过滤器是方便的。

步骤(1)给出过滤器中间值如下：

```
intermediate-filter  Filter ::=  and {
      item substrings  {
        type commonName,
        strings {any T61String "Dubois" }}, - value supplied for Common Name

      item substrings {
        type streetAddress
        strings {any T61String "Hugo" }}}   - value supplied for Street Address
```

步骤(2)给出一个过滤器，用它来寻找子树中匹配的住宅人员登录项：

```
residential-person-filter    Filter  ::=   {
  and {
      item equality {
         objectClass,
         OBJECT-CLASS residentialPerson },
      intermediate-filter }}
```

5.5.3 *Business Category* 职业种类

职业种类属性类型描述关于某些普通客体，如人的职业的情报。例如，此属性可做为向号码簿查询同行的手段。

```
businessCategory ATTRIBUTE
      WITH ATTRIBUTE-SYNTAX
            caseIgnoreStringSyntax
                  (SIZE(1..ub-business-category))
      ::=    {attributeType 15))
```

5.6 邮政寻址属性类型

这些属性类型涉及向一个客体进行物理邮政投递所需要的情报。

5.6.1 *Postal address* 邮政地址

邮政地址属性类型为邮政权威向被指名客体物理投递邮政信息提供所需要的地址信息。

邮政地址的属性值通常是由从建议 F.401 定义的 MHS 无格式邮政 O/R 地址第 1 版本中挑选的属性组成的，并规定最多为 6 行，每行为 30 个字符，包括一个邮政国别名。一般来说，在那样的地址里包含的信息，有可能包括收件人名称、街道地址、城市、州或省、邮政编码，可能还有邮政信箱号，视被指名客体的具体要求而定。

```
postalAddress ATTRIBUTE
      WITH ATTRIBUTE-SYNTAX PostalAddress
      MATCHES FOR EQUALITY
      ::=    {attributeType 16}

PostalAddress ::= SEQUENCE SIZE(1..ub-postal-line) OF CHOICE {
      T61String (SIZE(1..ub-postal-string)),
      PrintableString (SIZE(1..ub-postal-string))}
```

上述类型值的匹配规则与 caseIgnoreListSyntax 使用的规则一样。

5.6.2 *Postal Code* 邮政编码

邮政编码属性类型表示被指名客体的邮政编码。倘若此属性值存在，则它将构成客体的邮政地址的一部分。

邮政编码的属性值是一个字符串。

```
postalCode ATTRIBUTE
    WITH ATTRIBUTE-SYNTAX
        caseIgnoreStringSyntax
            (SIZE(1..ub-postal-code))
    ::=  {attributeType 17}
```

5.6.3 *Post Office Box* 邮政信箱

邮政信箱属性类型指定客体接收物理邮政投递使用的邮政信箱。若存在，则该属性值构成客体的邮政地址的一部分。

```
postOfficeBox ATTRIBUTE
    WITH ATTRIBUTE-SYNTAX
        caseIgnoreStringSyntax
            (SIZE(1..ub-post-office-box))
    ::= {attributeType 18}
```

5.6.4 *Physical Delivery Office Name* 物理投递局名

物理投递局名属性类型指出物理投递局所在的城市名称、村名等。

物理投递局名的属性值是一个字符串。

```
physicalDeliveryOfficeName ATTRIBUTE
    WITH ATTRIBUTE-SYNTAX
        caseIgnoreStringSyntax
            (SIZE(1..ub-physical-office-name))
    ::=  {attributeType 19}
```

5.7 电信寻址属性类型

这些属性类型涉及使用电信手段同客体进行通信所需要的地址信息。

5.7.1 *Telephone Number* 电话号码

电话号码属性类型指出同一个客体有关联的电话号码。

电话号码的属性值是一个字符串，它符合国际上统一采用的格式来表示国际电话号码，建议 E.123(例如："+44 582 10101")。

```
telephoneNumber ATTRIBUTE
    WITH ATTRIBUTE-SYNTAX
        telephoneNumberSyntax
    ::=  {attributeType 20}
```

5.7.2 *Telex Number* 用户电报号码

用户电报号码属性类型，表示一个同某一个客体有关联的用户电报终端的用户电报号码、国家代码和应答代码。

```
telexNumber ATTRIBUTE
    WITH ATTRIBUTE-SYNTAX TelexNumber
    ::=  {attributeType 21}
```

```
TelexNumber ::=    SEQUENCE{
     telexNumber          PrintableString,
          (SIZE(1..ub-telex-number)),
     countryCode          PrintableString,
          (SIZE(1..ub-country-code)),
     answerback           PrintableString,
          (SIZE(1..ub-answerback))}
```

5.7.3 *Teletex Terminal Identifier* 智能用户电报终端标识符

智能用户电报终端标识符属性类型表示与一个客体有关联的智能用户电报终端使用的智能用户电报终端标识符(和任选参数)。

智能用户电报终端标识符的属性值是一个字符串,该字符串符合 CCITT 建议 F.200,和建议 T.62 中的任选参数集。

```
teletexTerminalIdentifier ATTRIBUTE
     WITH ATTRIBUTE-SYNTAX
          TeletexTerminalIdentifier
     ::=    {attributeType 22}

TeletexTerminalIdentifier ::=   SEQUENCE {
     teletexTerminal        PrintableString
          (SIZE(1..ub-teletex-terminal-id)),
     parameters TeletexNonBasicParameters
          OPTIONAL}
```

5.7.4 *Facsimile Telephone Number* 传真电话号码

传真电话号码属性类型表示与一个客体有关联的传真终端使用的电话号码(和任选参数)。

传真电话号码的属性值是一个字符串,该字符串符合国际上统一采用的国际电话号码格式,建议 E.1xx (例如:"+81 3 347 7418")和一个任选字符串(根据建议 T.30 的格式)。

```
facsimileTelephoneNumber ATTRIBUTE
     WITH ATTRIBUTE-SYNTAX
          Facsimile TelephoneNumber
     ::= {attributeType 23}

FacsimileTelephoneNumber ::= SEQUENCE{
     telephoneNumber PrintableString
          (SIZE(1..ub-telephone-number)),
     parameters G3FacsimileNonBasicParameters
          OPTIONAL}
```

5.7.5 *X.121 Address X.121* 地址

X.121 地址属性类型表示与一个客体有关联的和由 CCITT 建议 X.121 定义的地址。

```
x121Address ATTRIBUTE
     WITH ATTRIBUTE-SYNTAX
          NumericString
               (SIZE(1..ub-x121-address))
     MATCHES FOR EQUALITY SUBSTRINGS
     ::= {attributeType 24}
```

这个类型的值使用的匹配规则与 **numeric String Syntax** 使用的一样。

5.7.6 *International ISDN Number* 国际 *ISDN* 编号

国际 *ISDN* 编号属性类型表示与一个客体有关联的国际 ISDN 编号。

国际 ISDN 编号的属性值是一个字符串，该字符串符合国际上统一采用和 CCITT 建议 E.164 定义的 ISDN 地址格式。

```
internationalISDNNumber ATTRIBUTE
    WITH ATTRIBUTE-SYNTAX
        NumericString
            (SIZE(1..ub-isdn-address))
    ::= {attributeType 25}
```

这个类型的值使用的匹配规则与 **numeric String Syntax** 使用的一样。

5.7.7 *Registered Address* 注册地址

注册地址属性类型表示与某一个城市的某个客体有关联的地址使用的助记符。助记符在该城市所在国家里注册，并应用于公众电报业务(根据建议 F.1)。

```
registeredAddress ATTRIBUTE
    WITH ATTRIBUTE-SYNTAX PostalAddress
    ::= {attributeType 26}.
```

5.7.8 *Destination Indicator* 目的地指示器

目的地指示器属性类型(根据建议 F.1 和 F.3)表示与提供公众电报业务所需要的客体(收件人)有关联的国家和城市。

目的地指示器属性值是一个字符串。

```
destinationIndicator ATTRIBUTE
    WITH ATTRIBUTE-SYNTAX
        PrintableString
            (SIZE(1..ub-destination-indicator))
    - alphabetical characters only
    MATCHES FOR EQUALITY SUBSTRINGS
    ::= {attributeType 27}
```

上述类型值的匹配规则与 **caseIgnore String Syntax** 使用的规则一样。

5.8 优先选择属性类型

这些属性类型涉及一个客体的优先选择。

5.8.1 *Preferred Delivery Method* 优先选择的投递方式

优先选择的投递方式属性类型表示客体的优先级，即与该客体进行通信时优先选择的通信方式。

```
preferredDeliveryMethod ATTRIBUTE
    WITH ATTRIBUTE-SYNTAX
        SEQUENCE OF INTEGER {
            any-delivery-method     (0),
            mhs-delivery            (1),
            physical-delivery       (2),
            telex-delivery          (3),
            teletex-delivery        (4),
            g3-facsimile-delivery   (5),
            g4-facsimile-delivery   (6),
            ia5-terminal-delivery   (7),
            videotex-delivery       (8),
            telephone-delivery      (9)}
    SINGLE VALUE

    ::= {attributeType 28}
```

5.9 *OSI 应用属性类型*

这些属性类型涉及在 OSI 应用层内的客体的相关情报。

5.9.1 *Presentation Address* 表示地址

表示地址属性类型表示与代表 OSI 应用实体的一个客体有关联的一个表示地址。
表示地址的属性值是一个在建议 X.200 中定义的表示地址。

```
presentationAddress ATTRIBUTE
    WITH ATTRIBUTE-SYNTAX
        PresentationAddress
    MATCHES FOR EQUALITY
    SINGLE VALUE
    ::= {attributeType 29}

PresentationAddress    ::=    SEQUENCE
    pSelector          [0]    OCTET STRING OPTIONAL,
    sSelector          [1]    OCTET STRING OPTIONAL,
    tSelector          [2]    OCTET STRING OPTIONAL,
    nAddresses         [3]    SET SIZE(1..MAX) OF OCTET STRING}
```

此类型值的匹配规则如下:给出的表示地址同保存的表示地址,只有当选择器相等,而且给出的 n 地址(nAddresses)是保存的表示地址的一个子集时才相匹配。

5.9.2 *Supported Application Context* 支持的应用上下文

支持的应用上下文属性类型表示客体(一个 OSI 应用实体)所支持的应用上下文的客体标识符。

```
supportedApplicationContext ATTRIBUTE
    WITH ATTRIBUTE-SYNTAX
        objectIdentifierSyntax
    ::=  {attributeType 30}
```

5.10 关系属性类型

这些属性类型涉及与特定客体保持某种关系的客体的有关情报。

5.10.1 *Member* 成员

成员属性类型表示与客体有关联的一组名称。
成员的属性值是一个可识别名。

```
member ATTRIBUTE
    WITH ATTRIBUTE-SYNTAX
        distinguishedNameSyntax
    ::=  {attributeType 31}
```

5.10.2 *Owner* 所有者

所有者属性类型表示一些客体的名称,这些客体对相关联的客体负有责任。
所有者的属性值是一个可识别名(它可以表示一组名称),并可以反复出现。

```
owner ATTRIBUTE
    WITH ATTRIBUTE-SYNTAX
        distinguishedNameSyntax
    ::=  {attributeType 32}
```

5.10.3 *Role Occupant* 角色扮演者

角色扮演者属性类型表示担任某一个组织的角色的客体名称。

角色扮演者的属性值是一个可识别名。

```
roleOccupant ATTRIBUTE
    WITH ATTRIBUTE-SYNTAX
        distinguishedNameSyntax
    ::=  {attributeType 33}
```

5.10.4 *See Also* 参见

参见属性类型表示其他号码簿客体的名称，该客体也许是同一个实在世界客体的另一些方面(在某种意义上)。

参见属性值是一个可识别名。

```
seeAlso ATTRIBUTE
    WITH ATTRIBUTE-SYNTAX
        distinguishedNameSyntax
    ::=  {attributeType 34}
```

5.11 安全保密属性类型

这些属性类型涉及某一个客体的安全保密或安全保密特权。这些属性类型，除客体标识符的分配之外，都在建议 X.509 中描述。

5.11.1 *User Password* 用户口令

```
userPassword UserPassword
    ::= {attributeType 35}
```

5.11.2 *User Certificate* 用户证件

```
userCertificate UserCertificate
    ::= {attributeType 36}
```

5.11.3 *CA Certificate CA* 证件

```
cACertificate CACertificate
    ::= {attributeType 37}
```

5.11.4 *Authority Revocation List* 权威撤销名单

```
authorityRevocationList AuthorityRevocationList
    ::= {attributeType 38}
```

5.11.5 *Certificate Revocation List* 证件撤销名单

```
certificateRevocationList CertificateRevocationList
    ::= {attributeType 39}
```

5.11.6 *Cross Certificate Pair* 交叉证件对

```
crossCertificatePair CrossCertificatePair
    ::= {attributeType 40}
```

第 2 部分 — 属性语法

6 属性语法定义

6.1 号码簿使用的属性语法

6.1.1 *Undefined* 未定义

ARCHIVES U.I.T. GENEVE

未定义属性语法是为一些属性提供的，该属性的值估计不会被号码簿拿去做比较。

为属性指定这种属性语法相当于为属性指定数据类型 **ANY**，而且在 **ATTRIBUTE** 宏里不提供匹配规则。

```
undefined ATTRIBUTE-SYNTAX
    ANY
    ::= {attributeSyntax 0}
```

6.1.2 *Distinguished Name* 可识别名

可识别名属性语法是为一些属性提供的，该属性的值是可识别名。这个语法，除客体标识符的分配之外，在建议 X.501 中定义。

```
distinguishedNameSyntax DistinguishedNameSyntax
    ::= {attributeSyntax 1}
```

6.1.3 *Object Identifier* 客体标识符

客体标识符属性语法是为一些属性提供的，该属性的值是客体标识符。这属性语法，除客体标识符的分配之外，在建议 X.501 中定义。

```
objectIdentifierSyntax  ObjectIdentifierSyntax
    ::= {attributeSyntax 2}
```

6.2 字符串属性语法

在 § 6.2.1 至 § 6.2.4 中描述的语法里，认为下列空格并不重要：

— 前空格(即位于第一个印字前的空格)；

— 后空格(即位于最后一个印字后的空格)；

— 连续的多个内部空格(把它们当做一个空格字符)。

符合这些语法的属性，在进行匹配时应忽略根据上述规则认为不重要的所有空格。

6.2.1 *Case Exact String* 字型准确字符串

字型准确字符串属性语法是为一些属性提供的，该属性的值是字符串(T.61 字符串或可大印字符串)，字型(大写或小写)在比较操作中是很重要的(例如，“Dundee”和“DUNDEE”是不匹配的)。

```
caseExactString ATTRIBUTE-SYNTAX
    CHOICE {T61String, PrintableString}
    MATCHES FOR EQUALITY SUBSTRINGS
    ::= {attributeSyntax 3}
```

具有这种语法的两个字符串彼此相等的条件是：字符串必须等长，对应的字符必须相同。一个可打印的字符串可以同 T.61 字符串相比较：若对应的字符都在可打印字符串的字符集中，则认为比较操作正常。但是，倘若 T.61 字符串中的字符不是在可打印字符串的字符集中，则比较操作宣告失败。

6.2.2 *Case Ignore String* 字型忽略字符串

字型忽略字符串属性语法是为一些属性提供的，该属性的值是字符串(T.61 字符串或可打印字符串)，但其中的字型(大写或小写)对比较操作而言并不重要(例如，“Dundee”和“DUNDEE”相匹配)。

```
caseIgnoreStringSyntax ATTRIBUTE-SYNTAX
    CHOICE {T61String, PrintableString}
    MATCHES FOR EQUALITY SUBSTRINGS
    ::= {attributeSyntax 4}
```

匹配规则与字型准确字符串属性语法使用的匹配规则相同，仅在字型上有差别的字符都被认为是相同的。

6.2.3 *Printable String* 可打印字符串

可打印字符串属性语法是为一些属性提供的，该属性的值是可打印的字符串。

```
printableStringSyntax ATTRIBUTE-SYNTAX
    PrintableString
    MATCHES FOR EQUALITY SUBSTRINGS
    ::= {attributeSyntax 5}
```

匹配规则与字型准确字符串属性语法使用的匹配规则相同。

6.2.4 *Numeric String* 数字字符串

数字字符串属性语法是为一些属性提供的，该属性的值是数字字符串。

```
numericStringSyntax ATTRIBUTE-SYNTAX
    NumericString
    MATCHES FOR EQUALITY SUBSTRINGS
    ::= {attributeSyntax 6}
```

匹配规则与字型准确字符串属性语法使用的匹配规则相同，唯在比较过程中所有的空格字符都省略。

6.2.5 *Case Ignore List* 字型忽略表

字型忽略表属性语法是为一些属性提供的，该属性的值是字符串序列(T.61 字符串或可打印字符串)，但在比较过程中，字型(大写或小写)是不重要的。

```
caseIgnoreListSyntax ATTRIBUTE-SYNTAX
    SEQUENCE OF
        CHOICE {T61String, PrintableString}
    MATCHES FOR EQUALITY SUBSTRINGS
    ::= {attributeSyntax 7}
```

两个字型忽略表相等的条件是：在每一个表中的字符串的数目相同，和对应的字符串相匹配。这个匹配规则与字型忽略字符串属性语法(§6.2.2)类似。

6.3 其他属性语法

6.3.1 *Boolean* 布尔

布尔属性语法是为一些属性提供的，该属性的值是布尔(即表示真或假)。

```
booleanSyntax ATTRIBUTE-SYNTAX
    BOOLEAN
    MATCHES FOR EQUALITY
    ::= {attributeSyntax 8}
```

这个语法的两个属性值相等的条件是：二者皆为真或假。

6.3.2 *Integer* 整数

整数属性语法是为一些属性提供的，该属性的值是整数。

```
integerSyntax ATTRIBUTE-SYNTAX
    INTEGER
    MATCHES FOR EQUALITY ORDERING
    ::= {attributeSyntax 9}
```

这个语法的两个属性值相等的条件是整数相同。整数的顺序必须符合顺序规则。

6.3.3 *Octet String* 八位组串

八位组串属性语法是为一些属性提供的，该属性的值是八位组串。

```
octetStringSyntax ATTRIBUTE-SYNTAX
    OCTET STRING
    MATCHES FOR EQUALITY SUBSTRINGS ORDERING
    ::= {attributeSyntax 10}
```

具有这个属性语法的两个串相匹配的条件是该两个串的长度相等，而且对应的八位组必须相同。为了确定串的顺序，必须从头开始对八位组串进行比较，找出有差别的一对，通过头两个八位组之间的顺序关系确定。

6.3.4 *UTC Time* 通用协调时间

通用协调时间属性语法是为一些属性提供的，该属性的值表示绝对时间。

```
uTCTimeSyntax ATTRIBUTE-SYNTAX
    UTCTime
    MATCHES FOR EQUALITY ORDERING
    ::= {attributeSyntax 11}
```

这个语法的两个属性值相等的条件是它们表示同一个时间。早些时间被认为是“少于”晚些时间。

6.3.5 *Telephone Number* 电话号码

电话号码属性语法是为一些属性提供的，该属性的值代表电话号码。

```
telephoneNumberSyntax ATTRIBUTE-SYNTAX
    PrintableString
        (SIZE{1..ub-telephone-number})
    MATCHES FOR EQUALITY SUBSTRINGS
    ::= {attributeSyntax 12}
```

匹配规则与字型准确属性语法使用的规则相同，不同的是所有的空格和“-”字符在比较过程中都可以忽略。

附 件 A

（附于建议 X.520）

使用 ASN.1 描述的属性类型集

本附件是本建议的一部分。

本附件包括本建议中出现的所有 ASN.1 类型和值定义，并以 ASN.1 模块 **SelectedAttributeTypes** 的形

式提供。

```
SelectedAttributeTypes      {joint-iso-ccitt ds(5) modules(1)
                            selectedAttributeTypes(5)}
DEFINITIONS ::=
BEGIN
-- Exports everything --
IMPORTS
    informationFramework, authenticationFramework, attributeType,
    upperBounds
    FROM UsefulDefinitions          {joint-ISO-CCITT ds(5) modules(1)
                                    usefulDefinitions(0) },
    ATTRIBUTE, ATTRIBUTE-SYNTAX, AttributeType, OBJECT-CLASS,
    ObjectClass, AliasedObjectName,
    DistinguishedNameSyntax, ObjectIdentifierSyntax
        FROM InformationFramework informationFramework
    G3FacsimileNonBasicParameters,
    TeletexNonBasicParameters
        FROM MTSAbstractService {joint-ISO-CCITT mhs-motis(6)
                                mts(3) modules(0) mts-abstract-service(1)}
    UserCertificate, CACertificate, CrossCertificatePair, CertificateRevocationList,
    AuthorityRevocationList, UserPassword
        FROM AuthenticationFramework, authenticationFramework
        ub-answerback,
        ub-common-name, ub-surname, ub-serial-number,
        ub-locality-name, ub-state-name,
        ub-street-address, ub-organization-name,
        ub-organizational-unit-name, ub-title,
        ub-description, ub-business-category, ub-postal-line,
        ub-postal-string, ub-postal-code, ub-post-office-box,
        ub-physical-office-name, ub-telex-number,
        ub-country-code, ub-teletex-terminal-id,
        ub-telephone-number, ub-x121-address,
        ub-international-isdn-number, ub-destination-indicator,
        ub-user-password
            FROM UpperBounds upperBounds;
-- attribute types --
objectClass ObjectClass ::= {attributeType 0}

aliasedObjectName AliasedObjectName ::= {attributeType 1}

knowledgeInformation ATTRIBUTE
    WITH ATTRIBUTE-SYNTAX caseIgnoreStringSyntax
    ::= {attributeType 2}

commonName ATTRIBUTE
    WITH ATTRIBUTE-SYNTAX
        caseIgnoreStringSyntax
            (SIZE(1..ub-common-name))
    ::= {attributeType 3}

surname ATTRIBUTE
    WITH ATTRIBUTE-SYNTAX
        caseIgnoreStringSyntax
            (SIZE(1..ub-surname))
    ::= {attributeType 4}

serialNumber ATTRIBUTE
    WITH ATTRIBUTE-SYNTAX
        printableStringSyntax
            (SIZE(1..ub-serial-number))
    ::= {attributeType 5}
```

```
countryName ATTRIBUTE
    WITH ATTRIBUTE-SYNTAX
        PrintableString (SIZE(2)) -- IS 3166 codes only
    MATCHES FOR EQUALITY
    SINGLE VALUE
    ::= {attributeType 6}

localityName ATTRIBUTE
    WITH ATTRIBUTE-SYNTAX
        caseIgnoreStringSyntax
            (SIZE(1..ub-locality-name))
    ::= {attributeType 7}

stateOrProvinceName ATTRIBUTE
    WITH ATTRIBUTE-SYNTAX
        caseIgnoreStringSyntax
            (SIZE(1..ub-state-name))
    ::= {attributeType 8}

streetAddress ATTRIBUTE
    WITH ATTRIBUTE-SYNTAX
        caseIgnoreStringSyntax
            (SIZE(1..ub-street-address))
    ::= {attributeType 9}

organizationName ATTRIBUTE
    WITH ATTRIBUTE-SYNTAX
        caseIgnoreStringSyntax
            (SIZE(1..ub-organization-Name))
    ::= {attributeType 10}

organizationalUnitName ATTRIBUTE
    WITH ATTRIBUTE-SYNTAX
        caseIgnoreStringSyntax
            (SIZE(1..ub-organizational-unit-name))
    ::= {attributeType 11}

title ATTRIBUTE
    WITH ATTRIBUTE-SYNTAX
        caseIgnoreStringSyntax
            (SIZE(1..ub-title))
    ::= {attributeType 12}

description ATTRIBUTE
    WITH ATTRIBUTE-SYNTAX
        caseIgnoreStringSyntax
            (SIZE(1..ub-description))
    ::= {attributeType 13}

searchGuide ATTRIBUTE
    WITH ATTRIBUTE-SYNTAX
        Criteria
    ::= {attributeType 14}

Guide ::= SET {
    objectClass [0] OBJECT-CLASS OPTIONAL,
        criteria  [1]  Criteria }

Criteria ::=
    CHOICE {
        type    [0]    CriteriaItem,
        and     [1]    SET OF Criteria
        or      [2]    SET OF Criteria
        not     [3]    Criteria}

CriteriaItem ::=
    CHOICE {
        equality            [0]  AttributeType
        substrings          [1]  AttributeType
        greaterOrEqual      [2]  AttributeType
        lessOrEqual         [3]  AttributeType
        approximateMatch    [4]  AttributeType
```

```
businessCategory ATTRIBUTE
    WITH ATTRIBUTE-SYNTAX
        caseIgnoreStringSyntax
            (SIZE(1..ub-business-category))
    ::=  {attributeType 15}

postalAddress ATTRIBUTE
    WITH ATTRIBUTE-SYNTAX PostalAddress
    MATCHES FOR EQUALITY
    ::= {attributeType 16}

PostalAddress ::= SEQUENCE SIZE(1..ub-postal-line) OF
    CHOICE {
        T61String (SIZE(1..ub-postal-string)),
        PrintableString (SIZE(1..ub-postal-string))}

postalCode ATTRIBUTE
    WITH ATTRIBUTE-SYNTAX
        caseIgnoreStringSyntax
            (SIZE(1..ub-postal-code))
    ::=  {attributeType 17}

postOfficeBox ATTRIBUTE
    WITH ATTRIBUTE-SYNTAX
        caseIgnoreStringSyntax
            (SIZE(1..ub-post-office-box))
    ::=  {attributeType 18}

physicalDeliveryOfficeName ATTRIBUTE
    WITH ATTRIBUTE-SYNTAX
        caseIgnoreStringSyntax
            (SIZE(1..ub-physical-office-name))
    ::=  {attributeType 19}

telephoneNumber ATTRIBUTE
    WITH ATTRIBUTE-SYNTAX
        telephoneNumberSyntax
    ::=  {attributeType 20}

telexNumber ATTRIBUTE
    WITH ATTRIBUTE-SYNTAX TelexNumber
    ::= {attributeType 21}

TelexNumber ::= SEQUENCE {
    telexNumber PrintableString
        (SIZE(1..ub-telex-number)),
    countryCode PrintableString,
        (SIZE(1..ub-country-code)),
    answerback PrintableString
        (SIZE(1..ub-answerback))}

teletexTerminalIdentifier ATTRIBUTE
    WITH ATTRIBUTE-SYNTAX
        TeletexTerminalIdentifier
    ::= {attributeType 22}

TeletexTerminalIdentifier ::= SEQUENCE {
    teletexTerminalPrintableString
        (SIZE(1..ub-teletex-terminal-id)),
    parameters TeletexNonBasicParameters
        OPTIONAL}
facsimileTelephoneNumber ATTRIBUTE
    WITH ATTRIBUTE-SYNTAX
        FacsimileTelephoneNumber
    ::= {attributeType 23}

FacsimileTelephoneNumber  ::=  SEQUENCE {
    telephoneNumber PrintableString
        (SIZE(1..ub-telephone-number)),
    parameters G3FacsimileNonBasicParameters OPTIONAL}
```

```
x121Address ATTRIBUTE
    WITH ATTRIBUTE-SYNTAX
        NumericString
            (SIZE(1..ub-x121-address))
    MATCHES FOR EQUALITY SUBSTRINGS
    ::= {attributeType 24}

internationalISDNNumber ATTRIBUTE
    WITH ATTRIBUTE-SYNTAX
        NumericString
            (SIZE(1..ub-isdn-address))
    ::= {attributeType 25}

registeredAddress ATTRIBUTE
    WITH ATTRIBUTE-SYNTAX PostalAddress
    ::= {attributeType 26}

destinationIndicator ATTRIBUTE
    WITH ATTRIBUTE-SYNTAX
        PrintableString
            (SIZE(1..ub-destination-indicator))
    - alphabetical characters only
    MATCHES FOR EQUALITY SUBSTRINGS
    ::= {attributeType 27}

preferredDeliveryMethod ATTRIBUTE
    WITH ATTRIBUTE-SYNTAX
        SEQUENCE OF INTEGER {
            any-delivery-method         (0),
            mhs-delivery                (1),
            physical-delivery           (2),
            telex-delivery              (3),
            teletex-delivery            (4),
            g3-facsimile-delivery       (5),
            g4-facsimile-delivery       (6),
            ia5-terminal-delivery       (7),
            videotex-delivery           (8),
            telephone-delivery          (9)}
    SINGLE VALUE
    ::= {attributeType 28}

presentationAddress ATTRIBUTE
    WITH ATTRIBUTE-SYNTAX
        PresentationAddress
    MATCHES FOR EQUALITY
    SINGLE VALUE
    ::= {attributeType 29}

PresentationAddress  ::=   SEQUENCE {
    pSelector        [0]   OCTET STRING OPTIONAL,
    sSelector        [1]   OCTET STRING OPTIONAL,
    tSelector        [2]   OCTET STRING OPTIONAL,
    nAddresses       [3]   SET SIZE(1..MAX) OF OCTET STRING}

supportedApplicationContext ATTRIBUTE
    WITH ATTRIBUTE-SYNTAX
        objectIdentifierSyntax
    ::=  {attributeType 30}

member ATTRIBUTE
    WITH ATTRIBUTE-SYNTAX
        distinguishedNameSyntax
    ::=  {attributeType 31}

owner ATTRIBUTE
    WITH ATTRIBUTE-SYNTAX
        distinguishedNameSyntax
    ::=  {attributeType 32}
```

```
roleOccupant ATTRIBUTE
    WITH ATTRIBUTE-SYNTAX
        distinguishedNameSyntax
    ::= {attributeType 33}

seeAlso ATTRIBUTE
    WITH ATTRIBUTE-SYNTAX
        distinguishedNameSyntax
    ::= {attributeType 34}

userPassword UserPassword
    ::= {attributeType 35}

userCertificate UserCertificate
    ::= {attributeType 36}

cACertificate CACertificate
    ::= {attributeType 37}

authorityRevocationList AuthorityRevocationList
    ::= {attributeType 38}

certificateRevocationList CertificateRevocationList
    ::= {attributeType 39}

CrossCertificatePair CrossCertificatePair
    ::= {attributeType 40}

-- attribute syntaxes --

undefined ATTRIBUTE-SYNTAX
    ANY
    ::= {attributeSyntax 0}

distinguishedNameSyntax DistinguishedNameSyntax
    ::= {attributeSyntax 1}

objectIdentifierSyntax ObjectIdentifierSyntax
    ::= {attributeSyntax 2}

caseExactStringSyntax ATTRIBUTE-SYNTAX
    CHOICE {T61String, PrintableString}
    MATCHES FOR EQUALITY SUBSTRINGS
    ::= {attributeSyntax 3}

caseIgnoreSyntax ATTRIBUTE-SYNTAX
    CHOICE {T61String, PrintableString}
    MATCHES FOR EQUALITY SUBSTRINGS
    ::= {atrributeSyntax 4}

printableStringSyntax ATTRIBUTE-SYNTAX
    PrintableString
    MATCHES FOR EQUALITY SUBSTRINGS
    ::= {attributeSyntax 5}

numericStringSyntax ATTRIBUTE-SYNTAX
    NumericString
    MATCHES FOR EQUALITY SUBSTRINGS
    ::= {attributeSyntax 6}

caseIgnoreListSyntax ATTRIBUTE-SYNTAX
    SEQUENCE OF
        CHOICE {T61String, PrintableString}
    MATCHES FOR EQUALITY SUBSTRINGS
    ::= {attributeSyntax 7}

booleanSyntax ATTRIBUTE-SYNTAX
    BOOLEAN
    MATCHES FOR EQUALITY
    ::= {attributeSyntax 8}
```

```
integerSyntax ATTRIBUTE-SYNTAX
    INTEGER
    MATCHES FOR EQUALITY ORDERING
    ::= {attributeSyntax 9}

octetStringSyntax ATTRIBUTE-SYNTAX
    OCTET STRING
    MATCHES FOR EQUALITY SUBSTRINGS ORDERING
    ::= {attributeSyntax 10}

uTCTimeSyntax ATTRIBUTE-SYNTAX
    UTCTime
    MATCHES FOR EQUALITY ORDERING
    ::= {attributeSyntax 11}

telephoneNumberSyntax ATTRIBUTE-SYNTAX
    PrintableString
        (SIZE(1..ub-telephone-number))
    MATCHES FOR EQUALITY SUBSTRINGS
    ::= {attributeSyntax 12}
```

附 件 B

（附于建议 X.520）

属性类型和语法索引

属性类型

别名客体名*	§5.1.2	地方名	§5.3.2
权威撤销名单	§5.11.4	成员	§5.10.1
职业种类	§5.5.3	客体类别*	§5.1.1
CA 证件	§5.11.3	组织名称	§5.4.1
证件撤销名单	§5.11.5	组织单位名称	§5.4.2
普通名称	§5.2.1	所有者	§5.10.2
国别名	§5.3.1	物理投递局名	§5.6.4
交叉证件对	§5.11.6	邮政信箱	§5.6.3
说明	§5.5.1	邮政地址	§5.6.1
目的地指示器	§5.7.8	邮政编码	§5.6.2
传真电话号码	§5.7.4	优先选择的投递方式	§5.8.1
国际 ISDN 编号	§5.7.6	表示地址	§5.9.1
知识信息	§5.1.3	注册地址	§5.7.7

* 号码簿本身所知道和使用的。

角色扮演者	§5.10.3	**属性语法**	
搜索指南	§5.5.2	布尔	§6.3.1
参见	§5.10.4	字型准确字符串	§6.2.1
系列号	§5.2.3	字型忽略表	§6.2.5
州或省名	§5.3.3	字型忽略字符串	§6.2.2
街道地址	§5.3.4	可识别名*	§6.1.2
支持的应用上下文	§5.9.2	整数	§6.3.2
姓	§5.2.2	数字字符串	§6.2.4
电话号码	§5.7.1	客体标识符*	§6.1.3
智能用户电报终端标识符	§5.7.3	八位组串	§6.3.3
用户电报号码	§5.7.2	可打印字符串	§6.2.3
头衔	§5.4.3	电话号码	§6.3.5
用户证件	§5.11.2	通用协调时间	§6.3.4
用户口令	§5.11.1	未定义	§6.1.1
X.121 地址	§5.7.5		

附 件 C

（附于建议 X.520）

上 限

本附件是本建议的一部分。

```
UpperBounds   {joint-ISO-CCITT ds(5) modules(1)
              upperBounds(10)}

DEFINITIONS ::=
BEGIN

-- Exports everything --

ub-answerback                    INTEGER ::= 8
ub-common-name                   INTEGER ::= 64
ub-surname                       INTEGER ::= 64
ub-serial-number                 INTEGER ::= 64
ub-locality-name                 INTEGER ::= 128
ub-state-name                    INTEGER ::= 128
ub-street-address                INTEGER ::= 128
ub-organization-name             INTEGER ::= 64
ub-organizational-unit-name      INTEGER ::= 64
ub-title                         INTEGER ::= 64
```

* 号码簿本身所知道和使用的。

```
        ub-description                      INTEGER ::= 1024
        ub-business-category                INTEGER ::= 128
        ub-postal-line                      INTEGER ::= 6
        ub-postal-string                    INTEGER ::= 30
        ub-postal-code                      INTEGER ::= 40
        ub-post-office-box                  INTEGER ::= 40
        ub-physical-office-name             INTEGER ::= 128
        ub-telex-number                     INTEGER ::= 14
        ub-country-code                     INTEGER ::= 4
        ub-teletex-terminal-id              INTEGER ::= 24
        ub-telephone-number                 INTEGER ::= 32
        ub-x121-address                     INTEGER ::= 15
        ub-international-isdn-number        INTEGER ::= 16
        ub-destination-indicator            INTEGER ::= 128
        ub-user-password                    INTEGER ::= 128
    END
```

建　　议　X.521

号码簿客体类别集[①]

（1988年订于墨尔本）

目　　录

0　绪论 …… 213

1　应用范围 …… 214

2　参考文献 …… 214

3　定义和缩写词 …… 214

　3.1　OSI参考模型定义 …… 214
　3.2　号码簿模型定义 …… 214

4　符号法 …… 214

第1部分—客体类别集 …… 215

① 建议X.521和ISO9594-7信息处理系统—开放系统互连—号码簿客体类别集选是在紧密合作下制定的，而且在技术上是协调的。

5 有用属性集的定义 …… 215

5.1 电信属性集 …… 215
5.2 邮政属性集 …… 215
5.3 地点属性集 …… 215
5.4 组织属性集 …… 215

6 客体类别集定义 …… 216

6.1 顶 …… 216
6.2 别名 …… 216
6.3 国别 …… 216
6.4 地方 …… 216
6.5 组织 …… 216
6.6 组织单位 …… 216
6.7 人员 …… 217
6.8 组织人员 …… 217
6.9 组织角色 …… 217
6.10 一组名称 …… 217
6.11 住宅人员 …… 218
6.12 应用进程 …… 218
6.13 应用实体 …… 218
6.14 DSA …… 219
6.15 装置 …… 219
6.16 强验证用户 …… 219
6.17 发证机构 …… 219

附件 *A* — 使用 ASN.1 描述的客体类别集 …… 220
附件 *B* — 推荐的名称形式和 DIT 结构 …… 223

0 绪论

0.1 本文和本系列其他文件的目的在于简化信息处理系统之间的互连，以便提供号码簿服务。全套系统，包括它们的号码簿信息，可视为一个整体，叫做号码簿。号码簿包含的全部信息叫做号码簿信息库(DIB)，其典型用途是简化客体之间以及同这些客体，或围绕这些客体(如应用实体、人员、终端和分布表)的通信联系。

0.2 号码簿在开放系统互连中扮演重要角色；开放系统的目的是除使用互连标准之外，设法只通过很少的技术协议，就能实现下列信息处理系统之间的互连：

— 不同厂家；
— 不同管理部门；
— 不同复杂程度；
— 不同年代。

0.3 本建议定义(在第 1 部分)一些在一系列号码簿应用中可能要用到的客体类别。

0.4 附件 A 是本建议的一部分，它提供一个 ASN.1 模块，此模块包含在本文件中出现的所有类型和值的定义。

0.5 附件 B 不是本建议的一部分，它提供一些普通命名和结构规则。行政权威可能使用也可能不使用。

1 应用范围

1.1 本建议定义一些在一系列号码簿应用中可能要用到的属性集和客体类别集。属性集的定义包括为它所包含的属性提供识别符和为定义客体类别提供方便。客体类别的定义包括给它分配一个客体标识符，并列出与该类别的客体有关的一些属性类型。这些定义被负责管理号码簿信息的行政权威所使用。

1.2 任何行政权威都可以为任何目的定义自己的客体类别和子类别。

注 1 — 这些定义可用于也可不用于建议 X.501 中定义的符号法。

注 2 — 凡是在语义学上适合于应用的场合，建议使用本文件定义的客体类别或由此类别派生的子类别，而不主张产生新的客体类别。

1.3 行政权威可支持它所选择的客体类别的一部分或全部，而且还可以增加客体类别。

所有的行政权威应支持号码簿为达到自己的目的所使用的客体类别(顶、别名和 DSA 客体类别)

2 参考文献

建议 X.200 — 开放系统互连基本参考模型(参见 ISO7498)
建议 X.500 — 号码簿概念、模型和服务概述(参见 ISO9594-1)
建议 X.501 — 号码簿模型(参见 ISO9594-2)。

3 定义和缩写词

3.1 *OSI 参考模型定义*

本建议使用建议 X.200 定义的下列术语：

a)*application-entity* 应用实体；
b)*application-process* 应用进程。

3.2 号码簿模型定义

本建议使用建议 X.501 定义的下列术语：

a) *attribute* 属性；
b) *attribute type* 属性类型；
c) *Directory Information Tree (DIT)* 号码簿信息树(DIT)；
d) *Directory System Agent (DSA)* 号码簿系统代理(DSA)；
e) *attribute set* 属性集；
f) *entry* 登录项；
g) *name* 名称；
h) *object class* 客体类别；
i) *subclass* 子类别。

4 符号法

客体类别在本文是用特殊符号定义的，即在建议 X.501 中定义为 ASN.1 宏，**OBJECT-CLASS** 的符号。“属”客体标识符(**objectClass**)是用来指定分配给客体类别的客体标识符。它的定义在建议 X.501 的附件 B 中给出。

属性集在本文是用特殊符号定义的，即在建议 X.501 中定义为 ASN.1 宏，**ATTRIBUTE-SET** 的符号。“属”客体标识符(**attributeSet**)是用来指定分配给属性集定义的客体标识符。它的定义在建议 X.501 的附件 B 中给出。

第 1 部分 — 客体类别集

5 有用属性集的定义

5.1 *Telecommunication Attribute Set* 电信属性集

本属性集是用来定义那些在商业通信中常用的属性。

```
telecommunicationAttributeSet ATTRIBUTE-SET
    CONTAINS {
        facsimileTelephoneNumber,
        iSDNAddress,
        telephoneNumber,
        teletexTerminalIdentifier,
        telexNumber, X121Address,
        preferredDeliveryMethod,
        destinationIndicator,
        registeredAddress}
        ::= {attributeSet 0}
```

5.2 *Postal Attribute Set* 邮政属性集

本属性集是用来定义那些直接与邮政投递有关的属性。

```
postalAttributeSet ATTRIBUTE-SET
    CONTAINS {
        physicalDeliveryOfficeName,
        postalAddress,
        postalCode,
        postOfficeBox,
        streetAddress}
        ::= {attributeSet 1}
```

5.3 *Locale Attribute Set* 地点属性集

本属性集是用来定义那些常用于搜索目的、并指出一个客体的所在地的属性。

```
localeAttributeSet ATTRIBUTE-SET
    CONTAINS {
        localityName,
        stateOrProvinceName,
        streetAddress}
        ::= {attributeSet 2}
```

5.4 *Organizational Attribute Set* 组织属性集

本属性集是用来定义一个组织或一个组织单位通常拥有的一些属性。

```
organizationalAttributeSet ATTRIBUTE-SET
    CONTAINS {
        description,
        localeAttributeSet,
        postalAttributeSet,
        telecommunicationAttributeSet,
        businessCategory,
        seeAlso,
        searchGuide,
        userPassword}
        ::= {attributeSet 3}
```

6 客体类别集定义

6.1 *Top* 项

顶客体类别(其他的客体类别是它的子类别)在建议 X.501 中定义,唯不包括客体标识符的分配。

```
top  Top ::= {objectClass 0}
```

6.2 Alias 别名

别名客体类别(从这个类别可以导出别名登录项的类别)在建议 X.501 中定义,唯不包括客体标识符的分配。

```
alias  Alias ::= {objectClass 1}
```

6.3 *Country* 国别

国别客体类别用来定义 DIT 中的国别登录项。

```
country OBJECT-CLASS
      SUBCLASS OF top
      MUST CONTAIN {
            countryName}
      MAY CONTAIN {
            description,
            searchGuide}
      ::= {objectClass 2}
```

6.4 *Locality* 地方

地方客体类别用来定义 DIT 中的地方。

```
locality OBJECT-CLASS
      SUBCLASS OF top
      MAY CONTAIN {
            description,
            localityName,
            stateOrProvinceName,
            searchGuide,
            seeAlso,
            streetAddress}
      ::= {objectClass 3}
```

至少必须有一个地方名、州名或省名。

6.5 *Organization* 组织

组织客体类别用来定义 DIT 中的组织登录项。

```
organization OBJECT-CLASS
      SUBCLASS OF top
      MUST CONTAIN {
            organizationName}
      MAY CONTAIN {
            organizationalAttributeSet}
      ::= {objectClass 4}
```

6.6 *Organizational Unit* 组织单位

组织单位客体类别用来定义代表分部门或组织的登录项。

```
organizationalUnit OBJECT-CLASS
    SUBCLASS OF top
    MUST CONTAIN {
        organizationalUnitName}
    MAY CONTAIN {
        organizationalAttributeSet}
    ::= {objectClass 5}
```

6.7 *Person* 人员

人员客体类别用来定义代表一般人的登录项。

```
person OBJECT-CLASS
    SUBCLASS OF top
    MUST CONTAIN {
        commonName,
        surname}
    MAY CONTAIN {
        description,
        seeAlso,
        telephoneNumber,
        userPassword}
    ::= {objectClass 6}
```

6.8 *Organizational Person* 组织人员

组织人员客体类别用来定义代表被某一个组织雇用或与某一个组织有某种重要关系的人员的登录项。

```
organizationalPerson OBJECT-CLASS
    SUBCLASS OF person
    MAY CONTAIN {
        localeAttributeSet,
        organizationalUnitName,
        postalAttributeSet,
        telecommunicationAttributeSet,
        title}
    ::= {objectClass 7}
```

6.9 *Organizational Role* 组织角色

组织角色客体类别用来定义代表某一个组织角色(即组织内部的一个职务或角色)的登录项。一个组织角色通常被认为是由一个特定组织人员所扮演的角色。但是,一个组织角色在它的有效寿命期间,也可能陆续被其他许多人员所填补。通常,一个组织角色可由一个人或非人实体扮演。

```
organizationalRole OBJECT-CLASS
    SUBCLASS OF top
    MUST CONTAIN {
        commonName}
    MAY CONTAIN {
        description,
        localeAttributeSet,
        organizationalUnitName,
        postalAttributeSet,
        preferredDeliveryMethod,
        roleOccupant,
        seeAlso,
        telecommunicationAttributeSet}
    ::= {objectClass 8}
```

6.10 *Group of Name* 一组名称

一组名称客体类别用来定义代表一组不排序名称的登录项,该名称代表个别客体或其他若干组名称。一个组的成员数是静态的,即成员数是通过行政行为改变的,而不是每次指定该组时动态确定的。

通过以成员代替每一个组的办法,可以把一个组的成员改变为一组个别客体名。这个过程可以反复进行,直到全部组名成分都消除而只留下个别客体名为止。

```
groupOfNames OBJECT-CLASS
    SUBCLASS OF top
    MUST CONTAIN {
        commonName,
        member}
    MAY CONTAIN {
        description,
        organizationName,
        organizationalUnitName,
        owner,
        seeAlso,
        businessCategory}
    ::= {objectClass 9}
```

6.11 *Residential Person* 住宅人员

住宅人员客体类别用来定义代表住宅环境里的人的登录项。

```
residentialPerson OBJECT-CLASS
    SUBCLASS OF person
    MUST CONTAIN {
        localityName}
    MAY CONTAIN {
        localeAttributeSet,
        postalAttributeSet,
        preferredDeliveryMethod,
        telecommunicationAttributeSet,
        businessCategory}
    ::= {objectClass 10}
```

6.12 *Application Process* 应用进程

应用进程客体类别用来定义代表应用进程的登录项。一个应用进程是一个实际开放系统内的一个元素，该元素为某一个应用完成信息处理(见建议 X.200)。

```
applicationProcess OBJECT-CLASS
    SUBCLASS OF top
    MUST CONTAIN {
        commonName}
    MAY CONTAIN {
        description,
        localityName,
        organizationalUnitName,
        seeAlso}
    ::= {objectClass 11}
```

6.13 *Application Entity* 应用实体

应用实体客体类型用来定义代表应用实体的一些登录项。一个应用实体是由属于 OSI 的一个应用进程的诸方面组成的。

```
applicationEntity OBJECT-CLASS
    SUBCLASS OF top
    MUST CONTAIN {
        commonName,
        presentationAddress}
    MAY CONTAIN {
        description,
        localityName,
```

```
        organizationName,
        organizationalUnitName,
        seeAlso,
        supportedApplicationContext}
    ::= {objectClass 12}
```

注 — 倘若应用实体是用一个不同于应用进程的号码簿客体表示，则应使用 **CommonName** 来携带应用实体限定符的值。

6.14 *DSA*

DSA 客体类别用来定义代表各 DSA 的登录项。DSA 在建议 X.501 中定义。

```
dSA OBJECT-CLASS
    SUBCLASS OF applicationEntity
    MAY CONTAIN {
        knowledgeInformation}
    ::= {objectClass 13}
```

6.15 *Device* 装置

装置客体类别用来定义代表各装置的登录项。一个装置是指能够进行通信的一个物理单元，如调制解调器、磁盘驱动器等。

```
device OBJECT-CLASS
    SUBCLASS OF top
    MUST CONTAIN {
        commonName}
    MAY CONTAIN {
        description,
        localityName,
        organizationName,
        organizationalUnitName,
        owner,
        seeAlso,
        serialNumber}
    ::= {objectClass 14}
```

注 — 至少包括 **localityName** ，**serialNumber**，**Owner** 之一，随选用的装置类型而定。

6.16 *Strong Authentication User* 强验证用户

强验证用户类别用来定义参与建议 X.509 定义的强验证的客体的登录项。

```
strongAuthenticationUser OBJECT-CLASS
    SUBCLASS OF top
    MUST CONTAIN {userCertificate}
    ::= {objectClass 15}
```

6.17 发证机构

发证机构客体类别用来定义担任如建议 X.509 所定义的发证机构的客体的登录项。

```
certificationAuthority OBJECT-CLASS
    SUBCLASS OF top
    MUST CONTAIN {
        cACertificate,
        certificateRevocationList,
        authorityRevocationList }
    MAY CONTAIN {crossCertificatePair}
    ::= {objectClass 16}
```

附 件 A

（附于建议 X.521）

使用 ASN.1 描述的客体类别集

本附件包括本建议中出现的所有 ASN.1 类型和值定义，并以 ASN.1 模块 **SelectedObjectClasses** 的形式提供。

```
SelectedObjectClasses    {joint-ISO-CCITT ds(5) modules(1)
                          selectedObjectClasses(6)}
DEFINITIONS ::=
BEGIN
-- exports everything
IMPORTS
    objectClass, attributeSet, informationFramework, selectedAttributeTypes
        FROM UsefulDefinitions {joint-iso-ccitt ds(5) modules(1) usefulDefinitions(0)}
    OBJECT-CLASS,ATTRIBUTE-SET, Top, Alias
        FROM InformationFramework informationFramework
    authorityRevocationList, businessCategory, CACertificate, certificateRevocationList,
    commonName, countryName, description, destinationIndicator, facsimileTelephoneNumber,
    internationalISDNNumber, knowledgeInformation, localityName, member, organizationName,
    organizationalUnitName, owner, physicalDeliveryOfficeName, postOfficeBox, postalAddress,
    postalCode, preferredDeliveryMethod, presentationAddress, registeredAddress,
    roleOccupant, searchGuide, seeAlso, serialNumber, stateOrProvinceName, streetAddress,
    supportedApplicationContext, surname, telephoneNumber, teletexTerminalIdentifier,
    telexNumber, title, userCertificate, userPassword, x121Address
        FROM SelectedAttributeTypes selectedAttributeTypes;
telecommunicationAttributeSet ATTRIBUTE-SET
    CONTAINS {
        facsimileTelephoneNumber,
        iSDNAddress,
        telephoneNumber,
        teletexTerminalIdentifier,
        telexNumber,
        x121Address, preferredDeliveryMethod, destinationIndicator,
        registeredAddress}
        ::= {attributeSet 0}
postalAttributeSet ATTRIBUTE-SET
    CONTAINS {
        physicalDeliveryOfficeName,
        postalAddress,
        postalCode,
        postOfficeBox,
        streetAddress}
        ::= {attributeSet 1}
localeAttributeSet ATTRIBUTE-SET
    CONTAINS {
        localityName,
        stateOrProvinceName,
        streetAddress}
        ::= {attributeSet 2}
organizationalAttributeSet ATTRIBUTE-SET
    CONTAINS {
        description,
        localeAttributeSet,
        postalAttributeSet,
        telecommunicationAttributeSet,
        businessCategory,
```

```
            seeAlso,
            searchGuide,
            userPassword}
            ::= {attributeSet 3}

top         Top ::= {objectclass 0}

alias       Alias ::= {objectClass 1}

country OBJECT-CLASS
      SUBCLASS OF top
      MUST CONTAIN {
            countryName}
      MAY CONTAIN {
            description,
            searchGuide}
      ::= {objectClass 2}

locality OBJECT-CLASS
      SUBCLASS OF top
      MAY CONTAIN {
            description,
            localityName,
            stateOrProvinceName,
            searchGuide,
            seeAlso,
            streetAddress}
      ::= {objectClass 3}

organization OBJECT-CLASS
      SUBCLASS OF top
      MUST CONTAIN {
            organizationName}
      MAY CONTAIN {
            organizationalAttributeSet}
      ::= {objectClass 4}

organizationalUnit OBJECT-CLASS
      SUBCLASS OF top
      MUST CONTAIN {
            organizationalUnitName}
      MAY CONTAIN {
            organizationalAttributeSet}
      ::= {objectClass 5}

person OBJECT-CLASS
      SUBCLASS OF top
      MUST CONTAIN {
            commonName,
            surname}
      MAY CONTAIN {
            description,
            seeAlso,
            telephoneNumber,
            userPassword}
      ::= {objectClass 6}

organizationalPerson OBJECT-CLASS
      SUBCLASS OF person
      MAY CONTAIN {
            localeAttributeSet,
            organizationalUnitName,
            postalAttributeSet,
            telecommunicationAttributeSet,
            title}
      ::= {objectClass 7}
```

```
organizationalRole OBJECT-CLASS
    SUBCLASS OF top
    MUST CONTAIN {
        commonName}
    MAY CONTAIN {
        description,
        localeAttributeSet,
        organizationalUnitName,
        postalAttributeSet,
        preferredDeliveryMethod,
        roleOccupant,
        seeAlso,
        telecommunicationAttributeSet}
    ::= {objectClass 8}

groupOfNames OBJECT-CLASS
    SUBCLASS OF top
    MUST CONTAIN {
        commonName,
        member}
    MAY CONTAIN {
        description,
        organizationName,
        organizationalUnitName,
        owner,
        seeAlso,
        businessCategory}
    ::= {objectClass 9}

residentialPerson OBJECT-CLASS
    SUBCLASS OF person
    MUST CONTAIN {
        localityName}
    MAY CONTAIN {
        localeAttributeSet,
        postalAttributeSet,
        preferredDeliveryMethod,
        telecommunicationAttributeSet,
        businessCategory}
    ::= {objectClass 10}

applicationProcess OBJECT-CLASS
    SUBCLASS OF top
    MUST CONTAIN {
        commonName}
    MAY CONTAIN {
        description,
        localityName,
        organizationalUnitName,
        seeAlso}
    ::= {objectClass 11}

applicationEntity OBJECT-CLASS
    SUBCLASS OF top
    MUST CONTAIN {
        commonName,
        presentationAddress}
    MAY CONTAIN {
        description,
        localityName,
        organizationName,
        organizationalUnitName,
        seeAlso,
        supportedApplicationContext}
    ::= {objectClass 12}
```

```
dSA OBJECT-CLASS
    SUBCLASS OF applicationEntity
    MAY CONTAIN {
        knowledgeInformation}
    ::= {objectClass 13}

device OBJECT-CLASS
    SUBCLASS OF top
    MUST CONTAIN {
        commonName}
    MAY CONTAIN {
        description,
        localityName,
        organizationName,
        organizationalUnitName,
        owner,
        seeAlso,
        serialNumber}
    ::= {objectClass 14}

strongAuthenticationUser OBJECT-CLASS
    SUBCLASS OF top
    MUST CONTAIN {
        userCertificate}
    ::= {objectClass 15}

certificationAuthority OBJECT-CLASS
    SUBCLASS OF top
    MUST CONTAIN {
        cACertificate,
        certificateRevocationList,
        authorityRevocationList}
    MAY CONTAIN {
        crossCertificatePair}
    ::=  {objectClass 16}

END
```

附 件 B

（附于建议 X.521）

推荐的名称形式和 DIT 结构

本附件不是本建议的一部分。

本附件介绍建议采用的一些通用命名方法和 DIT 结构，可供行政权威选用。客体类别的命名方法和 DIT 结构定义包括：用于命名的属性说明，以及说明在 DIT 中哪些客体类别是其上级登录项或下级登录项可以得到的。已知客体类别的全部登录项必须至少包含用于命名的属性。为了使号码簿用户能预告通信中使用的客体名，必须把建议采用的名称格式告诉用户。下面的章节为一些客体类别采用的命名和结构规则提出了建议。

结构规则示于图 B-1/X.521 中。

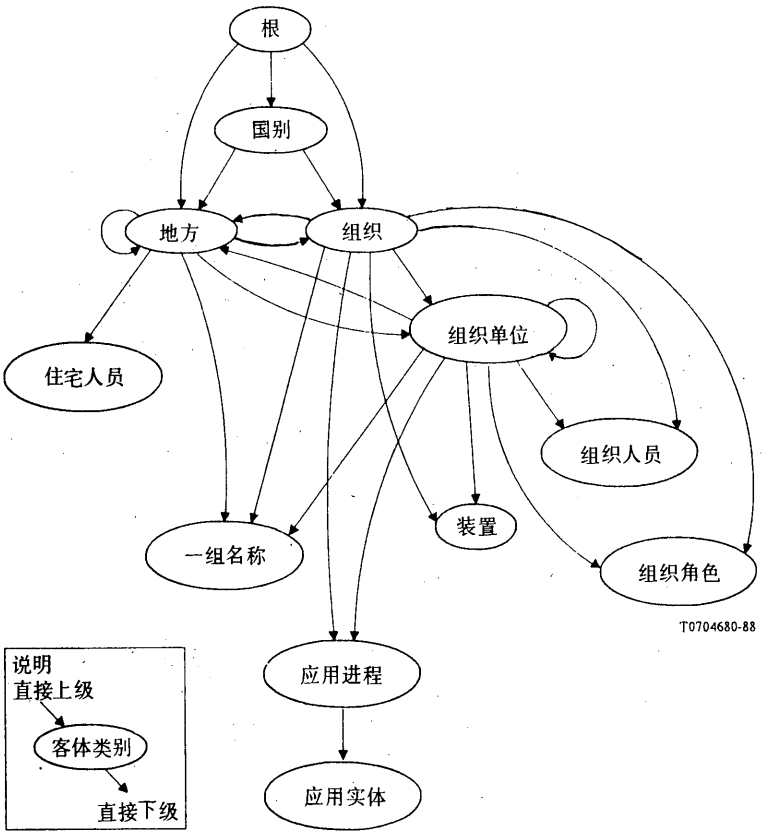


图 B-1/X. 521

建议采用的 DIT 结构

B. 1　国别

属性 **countryName** 用于命名目的。

根是客体类别**国别**的登录项的直接上级。

B. 2　组织

属性 **organizationName** 用于命名目的。

根、**国别**或**地方**都可以成为客体类别**组织**的登录项的直接上级。

注 — 当组织直接位于根的下面时，这表示一个国际组织。对于国际组织而言，所有的 **organizationName** 属性值都必须彼此不相同。

B. 3　地方

属性 **localityName** 或 **stateOrprovinceName** 用于命名目的。

根、**国别**、**地方**、**组织**或**组织单位**都可以成为客体类别 **locality** 的登录项的直接上级。

B. 4　组织单位

属性 **organizationalUnitName** 用于命名目的。

组织、**组织单位**或**地方**都可以成为客体类别 **organizationalUnit** 的登录项的直接上级。

B.5 组织人员

属性 **commonName** 或任选的 **organizationalUnitName** 用于命名目的。
组织或**组织单位**都可以成为客体类别 **organizationalPerson** 的登录项的直接上级。

注 — 在名称中，属性 **organizationalUnitName** 有两种获得的方法：即把**组织单位**客体当做上级，或直接获得这样的属性。

B.6 组织角色

属性 **CommonName** 用于命名目的。
组织或**组织单位**都可以成为客体类别 **organizationalRole** 的登录项的直接上级。

注 — 在名称中，属性 **organizationalUnitName** 有两种获得的方法：即把**组织单位**客体当做上级，或直接获得这样的属性。

B.7 一组名称

属性 **commonName** 是用于命名目的。
地方、**组织**、或**组织单位**都可以成为客体类别 **groupOfName** 的登录项的直接上级。

注 — 在名称中，属性 **organizationalUnitName** 有两种获得的方法：即把**组织单位**客体当做上级，或直接获得这样的属性。

B.8 住宅人员

属性 **commonName** 和任选的 **streetAddress** 用于命名目的。
地方是客体类别 **residentialPerson** 登录项的直接上级。

B.9 应用实体

属性 **commonName** 用于命名目的。**CommonName** 必须包含一个应用实体限定语(见建议 X.200)。
应用进程是客体类别 **applicationEntity** 登录项的直接上级。

B.10 装置

属性 **commonName** 用于命名目的。
组织或**组织单位**都可成为客体类别 **device** 登录项的直接上级。

注 — 在名称中，属性 **organizationalUnitName** 有两种获得的方法：即把**组织单位**客体当做上级，或直接获得这样的属性。

B.11 应用进程

属性 **commonName** 用于命名目的。
组织或**组织单位**都可成为客体类别 **applicationProcess** 登录项的直接上级。

注 1 — 建议 X.200 中详述了怎样为一个应用登录项选择 **commonName**。
注 2 — 在名称中，属性 **organizationalUnitName** 有两种获得的方法：即把**组织单位**客体当做上级，或直接获得这样的属性。

中国印刷 ISBN 92-61-03735-6