ICS 35.100
L 79

中华人民共和国国家标准

GB/T 16264.8—1996
idt ISO/IEC 9594-8:1990

信息技术　开放系统互连　目录
第8部分：鉴别框架

Information technology—Open systems interconnection—The directory Part 8: Authentication framework

1996-03-22发布　　1996-10-01实施

国家技术监督局　发布

中 华 人 民 共 和 国
国 家 标 准
信息技术 开放系统互连 目录
第8部分:鉴别框架
GB/T 16264.8—1996
*
中国标准出版社出版
北京复兴门外三里河北街16号
邮政编码:100045
电 话:68522112
中国标准出版社秦皇岛印刷厂印刷
新华书店北京发行所发行 各地新华书店经售

*
开本 880×1230 1/16 印张 2¼ 字数 63 千字
1997年3月第一版 1997年3月第一次印刷
印数 1—1 000
*
书号:155066·1-13557 定价 12.00 元
*
标 目 305—35

前　言

本标准等同采用国际标准 ISO/IEC 9594-8:1990《信息技术　开放系统互连目录　第 8 部分:鉴别框架》和 ISO/IEC 9594-8:1990/Cor.1:1991《信息技术　开发系统互连　目录　第 8 部分:鉴别框架技术修改 1》。

根据 ISO/IEC 9594-8:1990/Cor.1:1991,本标准对 7.2、7.6、9.4 和 C5.2 作了修改,并删去了 D2 章。

通过制定这项国家标准,以便为信息处理的目录服务提供统一的鉴别框架。

GB/T 16264 在《信息技术　开放系统互连　目录》总标题下,目前包括以下 8 个部分:

第 1 部分(即 GB/T 16264.1):概念、模型和服务的概述;

第 2 部分(即 GB/T 16264.2):模型;

第 3 部分(即 GB/T 16264.3):抽象服务定义;

第 4 部分(即 GB/T 16264.4):分布操作过程;

第 5 部分(即 GB/T 16264.5):协议规范;

第 6 部分(即 GB/T 16264.6):选择属性类型;

第 7 部分(即 GB/T 16264.7):选择客体类;

第 8 部分(即 GB/T 16264.8):鉴别框架。

本标准的附录 G 是标准的附录;

本标准的附录 A、B、C、D、E、F 和 H 是提示的附录。

本标准由中华人民共和国电子工业部提出。

本标准由电子工业部标准化研究所归口。

本标准起草单位:电子工业部标准化研究所、华北计算技术研究所。

本标准主要起草人:郑洪仁、李卫国、黄家英、冯惠。

ISO/IEC 前言

ISO(国际标准化组织)和 IEC(国际电工委员会)是世界性的标准化专门机构。国家成员体(它们都是 ISO 或 IEC 的成员国)通过国际组织建立的各个技术委员会参与制定针对特定技术范围的国际标准。ISO 和 IEC 的各技术委员会在共同感兴趣的领域内进行合作。与 ISO 和 IEC 有联系的其他官方和非官方国际组织也可参与国际标准的制定工作。

对于信息技术,ISO 和 IEC 建立了一个联合技术委员会,即 ISO/IEC JTC1。由联合技术委员会提出的国际标准草案需分发给国家成员体进行表决。发布一项国际标准至少需要 75%的参与表决的国家成员体投票赞成。

国际标准 ISO/IEC 9594.8 是由 ISO/IEC JTC1"信息技术"联合技术委员会制定的。

ISO/IEC 9594 在《信息技术　开放系统互连　目录》总标题下,目前包括以下 8 个部分:

——第 1 部分:概念、模型和服务的概述

——第 2 部分:模型

——第 3 部分:抽象服务定义

——第 4 部分:分布式操作规程

——第 5 部分:协议规范

——第 6 部分:选择属性类型

——第 7 部分:选择客体类

——第 8 部分:鉴别框架

附录 G 构成为 ISO/IEC 9594.8 的一部分,而附录 A、B、C、D、E、F 和 H 仅提供参考信息。

引　　言

0.1　本标准，连同本系列标准的其他几部分一起，便于提供目录服务的信息处理系统的互连。所有这样的系统连同它们所拥有的目录信息，可以看作一个整体，称为“目录”。目录中收录的信息在总体上称为目录信息库(DIB)，它可用于简化诸如OSI应用实体、人、终端，以及分布列表等客体之间的通信。

0.2　目录在开放系统互连中起着极其重要的作用，其目的是允许在互连标准之外使用最少的技术协定，完成下列各类信息处理系统的互连：

- 来自不同厂家的信息处理系统；
- 处在不同机构的信息处理系统；
- 具有不同复杂程度的信息处理系统；
- 不同年代的信息处理系统。

0.3　许多应用都有保护信息的通信免受威胁的安全要求。附录A概要描述了一些常见的威胁以及可用于保护信息免受这些威胁的安全服务和安全机制。实际上，所有的安全服务都依赖于通信各方的身份被可靠地认知，即，鉴别。

0.4　本标准借助目录给其用户的鉴别服务定义了一个框架。这些用户不仅包括其他应用和服务，还包括目录本身。目录常用来满足鉴别和其他安全服务的需要，因为目录是通信各方获得作为相互鉴别的基础的鉴别信息的一个自然场所；同时，在目录中还保存了用于满足通信请求并且在通信发生之前必须获得的其他信息。用这种方法还可以从目录中获得一个潜在通信伙伴的鉴别信息，类似于获得一个地址。由于以通信为目的的目录的适用范围很广泛，可以预期，许多应用都将广泛地使用这个鉴别框架。

目　　次

中华人民共和国国家标准

信息技术　开放系统互连　目录
第8部分：鉴别框架

GB/T 16264.8—1996
idt ISO/IEC 9594-8:1990

Information technology—Open systems interconnection—The directory Part 8:Authentication framework

第一篇　综　述

1　范围

1.1　本标准：

- 具体说明了目录拥有的鉴别信息的格式；
- 描述如何从目录中获得鉴别信息；
- 说明如何在目录中构成和存放鉴别信息的假设；
- 定义各种应用使用该鉴别信息执行鉴别的三种方法，并描述鉴别如何支持其他安全服务。

1.2　本标准描述了两级鉴别：简单鉴别，使用口令作为自称身份的一个验证；强鉴别，包括使用密码技术形成凭证。简单鉴别只提供一些有限的保护，以避免非授权的访问，只有强鉴别才可用作提供安全服务的基础。本标准不准备为鉴别建立一个通用框架，但本标准对于认为那些技术已经足够的应用来说可能是通用的，因为这些技术对它们已经足够了。

1.3　在一个已定义的安全策略上下文中仅提供鉴别(和其他安全服务)。因标准提供的服务而受限制的用户安全策略，由一个应用的用户自己来定义。

1.4　由使用本鉴别框架定义的应用的标准来指定必须执行的协议交换，以便根据从目录中获取的鉴别信息来完成鉴别。应用从目录中获取凭证的协议称作目录访问协议(DAP)，由 GB/T 16264.5 规定。

1.5　本标准中规定的强鉴别方法以公开密钥密码体制为基础。这种体制的主要优点是可以将用户证书作为目录的属性保存在目录中，并允许在目录系统中自由交换，目录的用户也可以采用与获取其他目录信息同样的方法获取用户证书。用户证书可以采用'脱机'方式形成，并由其创建者置入目录中。用户证书的生成应由完全独立于目录中的任何 DSA 的'证明职能机构'负责。尤其是，不应对目录提供者存储或交换用户证书所采用的安全方法作特殊的要求。

附录 B 给出了公开密钥密码的概要介绍。

1.6　在通常情况下，鉴别框架应独立于所采用的具有 6.1 所描述的特性的某种加密算法，也就是说，可以采用多种不同的加密算法。然而，想要相互鉴别的两个用户则支持采用相同的加密算法，从而确保进行正确的鉴别。因此，在一组相关的应用的上下文中，选择一种单一的算法将会增强用户进行安全鉴别和通信的一致性。

附录 C 给出了公开密钥加密算法的一个示例。

1.7　同样，想要相互鉴别的两个用户必须支持相同的散列函数(见 3.3.6)，散列函数主要用于生成凭证和鉴别权标。同样，从原理上说，也可以采用多种散列函数，但这要以减小用户鉴别的一致性为代价。

附录 D 给出了散列函数的概要介绍及示例。

国家技术监督局1996-03-22批准　　1996-10-01实施

2 引用标准

下列标准所包含的条文，通过在本标准中引用而构成为本标准的条文。本标准出版时，所示版本均为有效。所有标准都会被修订，使用本标准的各方应探讨使用下列标准最新版本的可能性。

GB/T 9387.2—1995 信息处理系统 开放系统互连 基本参考模型 第2部分：安全体系结构(idt ISO 7498-2:1987)

GB/T 16264.2—1996 信息技术 开放系统互连 目录 第2部分：模型(idt ISO/IEC 9594-2:1990)

GB/T 16264.3—1996 信息技术 开放系统互连 目录 第3部分：抽象服务定义(idt ISO/IEC 9594-3:1990)

GB/T 16264.4—1996 信息技术 开放系统互连 目录 第4部分：分布式操作规程(idt ISO/IEC 9594-4:1990)

GB/T 16264.5—1996 信息技术 开放系统互连 目录 第5部分：协议规范(idt ISO/IEC 9594-5:1990)

GB/T 16263—1996 信息技术 开放系统互连 用于抽象语法记法一(ASN.1)的基本编码规则规范(idt ISO/IEC 8825:1990)

3 定义

3.1 本标准使用在GB/T 9387.2中定义的有关安全的通用术语：

a) 不对称(加密) asymmetric(encipherment)；

b) 鉴别交换 authentication exchange；

c) 鉴别信息 authentication information；

d) 机密性 confidentiality；

e) 凭证 credentials；

f) 密码 cryptography；

g) 数据原发鉴别 data origin authentication；

h) 解密 decipherment；

i) 加密 encipherment；

j) 密钥 key；

k) 口令 password；

l) 对等实体鉴别 peer—entity authentication；

m) 对称(加密) symmetric(encipherment)。

3.2 本标准使用下列在GB/T 16264.2中定义的术语：

a) 属性 attribute；

b) 目录信息库 Directory Information Base；

c) 目录信息树 Directory Information Tree；

d) 可辨别名 distinguished name；

e) 项 entry；

f) 客体 object；

g) 根 root。

3.3 本标准还定义并使用下列术语：

3.3.1 鉴别权标(权标) authentication token(token)

在强鉴别交换期间运送的信息，可被用于鉴别其发送者。

3.3.2 用户证书(证书) user certificate(certificate)

用户的公开密钥,以及某些其他信息;与证明职能机构发出的秘密密钥一起用于数据加密。

3.3.3 证明职能机构(CA) certification authority

受一个或多个用户信任的职能机构,负责创建和分发证书。证明职能机构也可有选择地创建用户密钥。

3.3.4 证明路径 certification path

DIT 中客体证书的有序集合,可通过路径中的起始客体的公开密钥获取路径中终止客体的公开密钥。

3.3.5 密码体制 cryptographic system、cryptosystem

实现从明文到密文和从密文到明文的一组变换,所用的某个特定的变换由密钥选定。通常用一个数学算法来定义这种变换。

3.3.6 散列函数 hash function

将值从一个大的(可能很大)区域映射到一个小的区域的一个(数学)函数。一个"好的"散列函数是指该函数的结果应能均匀地(而且随机地)分布在由函数生成的值域中。

3.3.7 单向函数 one—way function

一个易于计算的(数学)函数,但对于域中的一个普通值 y 来说,要找到一个满足函数 $f(x)=y$ 的值 x 却相当困难。也许存在某些值 y,要找到其对应的 x 却不难。

3.3.8 公开密钥 public key

(在公开密钥密码体制中,)用户密钥对中为公众所知的那个密钥。

3.3.9 秘密密钥 private key

(在公开密钥密码体制中)用户密钥对中只由该用户知道的那个密钥。

3.3.10 简单鉴别 simple authentication

采用简单口令分配方法进行的鉴别。

3.3.11 安全策略 security policy

由管理使用和提供安全服务和设施的安全职能机构设立的一组规则。

3.3.12 强鉴别 strong authentication

采用密码派生凭证方法进行的鉴别。

3.3.13 信任 trust

当一个实体(第一个实体)假定另一个实体(第二个实体)完全按照它的期望进行动作时,则称第一个实体"信任"第二个实体。这种"信任"可能只适用于某些特殊功能。在鉴别框架中"信任"的主要任务是描述鉴别实体和证明职能机构之间的关系;一个鉴别实体必须能够肯定它可以"信任"证明职能机构创建唯一有效而且可信的证书。

3.3.14 证书序号 certificate serial number

一个由 CA 产生的唯一与证书相关的整数。

4 记法和缩略语

4.1 本标准中使用的记法见下面表 1 定义。

注:在该表中出现的符号 X、X_1、X_2 等分别代表用户名,符号 I 则代表任意信息。

4.2 本标准使用以下缩略语:

CA	证明职能机构
DIB	目录信息库
DIT	目录信息树
PKCS	公开密钥密码体制

表 1 记法

记法	涵 义
X_P	用户 X 的公开密钥
X_S	用户 X 的秘密密钥
$X_P〔I〕$	用 X 的公开密钥，对信息 I 进行加密
$X_S〔I〕$	用 X 的秘密密钥，对信息 I 进行加密
X{I}	由用户 X 为信息 I 加标记，它包含信息 I 和附加加密摘要
CA(X)	用户 X 的证明职能机构
$CA^n(X)$	(这里，$n>1$)：CA(CA(..n 次..(X)))
$X_1《X_2》$	由证明职能机构 X_1 发出的用户 X_2 的证书
$X_1《X_2》X_2《X_3》$	一个(任意长度的)证书链，其中每一项都是一个证书，并且其证明职能机构产生下一个证书。上式等价于下一个证书 $X_1《X_{n+1}》$。例如：A《B》B《C》提供与 A《C》相同的能力，即给定 A_P，可以从中找到 C_P
$X_{1P}\cdot X_1《X_2》$	一个证书(或证书链)的拆封操作，以便从中获得一个公开密钥。这是一个非确定操作，其左操作数为一个证明职能机构的公开密钥，右操作数则为该证明职能机构发出的一个证书。输出结果为用户的公开密钥，它们的证书为右操作数。例如：$A_P\cdot A《B》B《C》$指出一个操作，该操作使用 A 的公开密钥，从 B 的证书中获得 B 的公开密钥 B_P，然后再通过 B_P 来解封 C 的证书。操作的最终结果即为 C 的公开密钥 C_P
A→B	A 到 B 的证书路径由一个证书链构成，链的开端为 $CA(A)《CA^2(A)》$，末端为 CA(B)《B》

第二篇 简单鉴别

5 简单鉴别规程

5.1 简单鉴别提供建立在用户可辨别名、双方同意的(可选)口令、以及在某个单一区域中双方都能理解的口令的使用和处理方法之上的本地鉴别。简单鉴别一般只用于本地的对等实体，即一个 DUA 和一个 DSA 之间、或一个 DSA 与另一个 DSA 之间的鉴别。通常可采用以下几种方法实现简单鉴别：

a) 以清楚明确(即无保护)的方法将用户的可辨别名和(可选的)口令传送给接收方，以待考察；

b) 将用户的可辨别名、口令、以及一个随机数和/或时间标记通过使用单向函数进行保护，并传送；

c) 将 b)中描述的保护信息连同一个随机数和/或时间标记一起通过使用单向函数进行保护，并再传送。

注

1 不要求一定使用不同的单向函数。

2 用于保护口令的过程可能须要对本文件进行扩充。

5.2 如果口令没有被保护，则须提供最低限度的安全保护以防止未授权的访问；但不能将其看成是安全服务的基础。对用户的可辨别名和口令的保护则提供更高一级的安全。这里用于保护机制的算法都是一些典型的、极易实现的非加密单向函数。

5.3 图 1 给出了进行简单鉴别的一般过程。

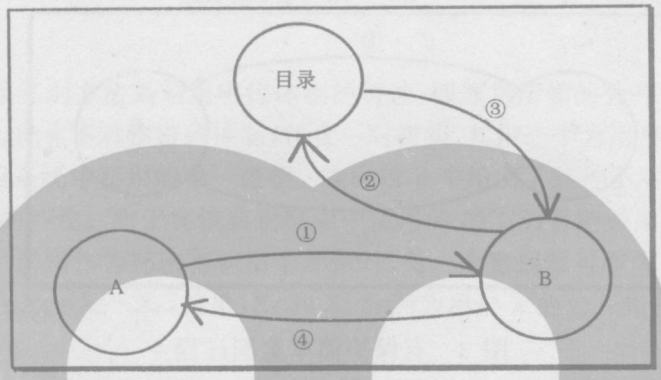

图 1 无保护的简单鉴别过程

5.3.1 简单鉴别过程一般包括以下几个步骤：

① 发方用户 A 将其可辨别名和口令发送给收方用户 B；

② 用户 B 将用户 A 声称的可辨别名和口令发送给目录，然后使用目录的比较操作对照检查与用户 A 有关的目录项的用户口令；

③ 目录向用户 B 返回是确认(或否认)该凭证是否有效的信息；

④ 可以向用户 A 运送鉴别的结果，即成功或失败信息。

5.3.2 简单鉴别的最基本的形式只包含步骤①，或在用户 B 检查完可辨别名和口令之后，也可包含步骤④。

5.4 图 2 用于保护可能产生的证明信息的两种方法。其中，$f1$ 和 $f2$ 为单向函数(它们可以相同，也可以不同)，并且时间标记和随机数为可选项，并服从双方约定。

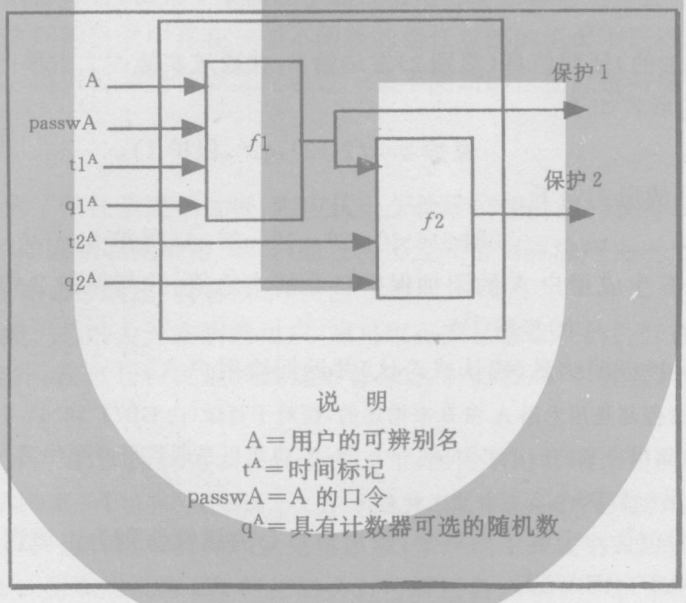

图 2 有保护简单鉴别

5.4.1 图 3 给出了有保护简单鉴别的过程。

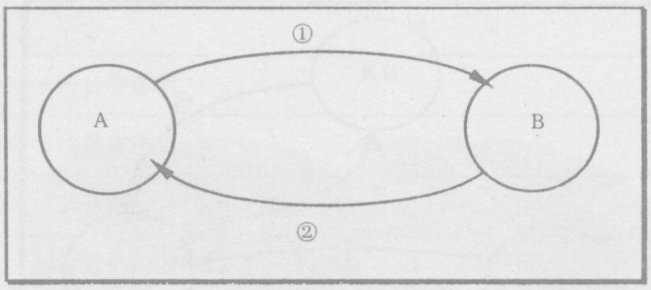

图 3 有保护简单鉴别过程

简单鉴别过程包含以下步骤(最初只使用 $f1$):

① 发方用户 A,向用户 B 发送其已保护的证明信息(鉴别 1)。并通过实施图 2 中的单向函数 $f1$ 进行保护,这里时间标记和/或随机数(当使用时)主要用于减少重复操作和隐藏口令。

A 的口令的保护形式如下:

$$保护\ 1=f1(t1^A,q1^A,A,passwA)。$$

运送给 B 的信息的形式如下:

$$鉴别\ 1=t1^A,q1^A,A,保护\ 1。$$

用户 B 通过使用 A 所提供的可辨别名、以及有选择的附加时间标记和/或随机数生成保护 1 形式的 A 的口令的一个本地副本来对 A 提供的保护证明信息进行核对,并将此证明信息(保护 1)与本地生成的值进行比较,检查是否相等。

② 用户 B 核对已保护证明信息(确认或否认)并返回给用户 A。

5.4.2 可对 5.4.1 所描述的鉴别过程进行修改以适应更高一级的鉴别(使用 $f1$ 和 $f2$)。

主要区别如下:

① 用户 A 将其(附加的)证明信息(鉴别 2)发送给 B;并通过实施图 2 中单向函数 $f2$ 来进行附加保护。附加保护的形式如下:

$$保护\ 2=f2(t2^A,q2^A,保护\ 1)。$$

运送给 B 的信息的形式如下:

$$鉴别\ 2=t1^A,t2^A,q1^A,q2^A,A,保护\ 2。$$

为了比较,用户 B 生成用户 A 的附加保护口令的本地值,并与保护 2 中的口令进行比较,检查是否相等(类似于 5.4.1 的步骤①)。

② 用户 B 核对已保护证明信息(确认或否认)并返回给用户 A。

注:在本章中定义的过程都是用术语 A 和 B 来描述的,而对于目录(由 GB/T 16264.3 和 GB/T 16264.4 规定的)来说,A 可以是联到用户 B 的一个 DSA 的一个 DUA,也可以是联到用户 B 的另一个 DSA 的一个 DSA,而 B 则是与 A 相对的另一个 DSA。

5.5 用户口令属性类型包含一个客体的口令;该用户口令的属性值则为由该客体指定的一个串。

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

5.6 下面 ASN.1 宏可以用来定义一个数据类型,而这个数据类型是由对某个给定的数据类型实施一个单向函数而生成的。

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

第三篇 强 鉴 别

6 强鉴别基础

6.1 本标准中采用的强鉴别方法是利用密码体制的特性，即通常所知的公开密钥密码体制(PKCS)来实现的。这种密码体制，也称不对称密码体制，包含一对密钥，其中一个为用户私有的，而另一个则为公开的；它不同于传统密码系统中使用的单一密钥。在附录 B 中给出了这种不对称密码体制的概要介绍，以及在鉴别过程中使用的特性。对于在该鉴别框架中使用的 PKCS，目前必须具备这样的特性，即在密钥对中，当其中任何一个密钥作为秘密密钥用于加密时，另一个密钥都可作为公开密钥用于解密；换句话说，它们必须满足 $X_P \cdot X_S = X_S \cdot X_P$，这里，$X_P$ 和 X_S 分别为用户 X 的公开和秘密密钥的加密和解密函数。

注：作为今后的扩充，PKCS 的替换类型不应要求具有可置换特性，并能在不对本标准作大的修改的情况下得到支持。

6.2 该鉴别框架并不强制使用某个特殊的密码体制，它适用于任何公开密码体制，并能支持今后对密码技术、数学技术、或可计算能力的更新。然而两个想要相互鉴别的用户必须支持相同的密码算法，从而能正确地执行鉴别。因此，在一组相关应用的上下文中，选择一个单一的算法，可以最大可能地增加用户间相互鉴别和安全通信的能力。附录 C 给出了密码算法的一个示例。

6.3 鉴别取决于每个具有唯一可辨别名的用户。可辨别名的分配由命名职能机构负责。每一个用户都应相信命名职能机构不会发出重复的可辨别名。

6.4 每个用户都可用其所拥有的秘密密钥来标识。另一个用户则可根据其通信对方是否拥有这个秘密密钥来确定他是否确实为(授权)用户。这种证实方法的有效性取决于只有用户才拥有该秘密密钥。

6.5 一个用户若要确定其通信对方是否拥有其他用户的秘密密钥，他自己就必须拥有该用户的公开密钥。用户的公开密钥的值可以直接从目录的用户项中获得，但要验证其正确性却有一定的问题。有许多可能的方法来验证用户的公开密钥：第 7 章描述了通过引用目录来验证用户公开密钥的操作过程。该过程只在请求鉴别的用户之间的目录中存在一条不间断的信任点链的情况下进行。这样的链可通过标记一个公共信任点来构造。而该公共信任点又应通过一条不间断的信任点链与每个用户相连。

7 用户公开密钥的获得

7.1 对一个用户来说，为了信任该鉴别过程，则应从一个他能信任的机构获得其他用户的公开密钥。这个可被公众信任的机构，即证明职能机构(CA)，通过使用公开密钥算法产生一个证书，并对公开密钥给予证明。证书(其格式在 7.2 中规定)具有以下特性：

- 任何访问证明职能机构的公开密钥的用户，都可以得到已被证明的公开密钥；
- 除证明职能机构外，没有任何其他组织能够修改这个证书，而不被查出(因为证书是不可伪造的)。

由于证书本身的不可伪造性，因此可以在目录中公布，而无需对目录作任何要求以保护这些证书。

注：尽管在 DIT 中，CA 是用一个可辨别名明确定义的，但这并不意味着在 CA 的组织和 DIT 之间存在任何联系。

7.2 证明职能机构通过对一组信息进行签名(见第 8 章)来产生用户证书，这些信息包括用户的可辨别名和公开密钥。例如，由证明职能机构 CA 产生的具有可辨别名 A 的用户证书的形式如下：

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

这里，SN 为证书序号，AI 为用于对证书进行签名的算法的标识符(该标识符与 SIGNED MACRO 值记法中规定的一致)，T^A 指出证书的有效期，它包含两个日期，只有当处于这两个日期之间时方才有效。T^A 的取值范围不少于 24 h，因此，希望系统使用国际标准时间(Coordinated Universal Time)作为参照的基础。证书中的签名的有效性可被具有 CA_P 知识的任何用户检查。证书可由以下 ASN.1 数据类型表示：

```
Certificate      ::=   SIGNED SEQUENCE{
    version                                  [0]  Version DEFAULT 1988,
    serialNumber                             CertificateSerialNumber,
    signature                                AlgorithmIdentifier,
    issuer                                   Name,
    validity                                 Validity,
    subject                                  Name,
    subjectPublicKeyInfo                     SubjectPublicKeyInfo}
Version      ::=      INTEGER{1988(0)}
CertificateSerialNumber      ::=      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 参与强鉴别的每个用户(例如,用户 A)的目录项,都包含 A 的证书。这样的证书一般由 A 的证明职能机构生成,而 A 的证明职能机构则为 DIT 的一个实体。A 的职能机构不必是唯一的,通常表记为 CA(A),或当 A 确定时,记为 CA。这样,A 的公开密钥可被任何知道 CA 的公开密钥的用户发现并得到。公开密钥是可重复发现的。

7.4 如果用户 A 已经获得用户 B 的证明职能机构 CA(B)的公开密钥,则获取用户 B 的公开密钥的操作结束。为使用户 A 能够获得 CA(B)的公开密钥,则每个证明职能机构 X 的目录项都包含许多证书。这些证书包括两种类型,一种是由其他证明职能机构生成的 X 的转发证书。另一种是由 X 自己生成的反向证书,它是其他证明职能机构的已证明的公开密钥。这些证书的存在使得用户能够构造从一点到另一点的证明路径。

7.5 需要有一个证书列表,以便一个特定用户可以获得其他用户的公开密钥,这就是所说的证明路径。在该列表中,每一项都是其后一项的证明职能机构的一个证书。从 A 到 B 的证明路径(表记 A→B)为:

- 由 CA(A)产生的一个证书开始,对应某个实体 X^1,命名为 CA(A)《X^1》;
- 依次由证书 X^f《X^{f+1}》继续;
- 到用户 B 的证书结束。

一个证明路径逻辑上在目录信息树中的两个想要相互鉴别的用户之间形成一条不间断的信任点链。用户 A 和用户 B 为获得证明路径 A→B 和 B→A 而采用的方法可以不同。可以采用一种简化途径,即将 CA 按层次安排,这种层次可以全部或部分与 DIT 的层次一致。这样做的好处是:在该层次结构中具有 CA 的用户可以不必了解任何其他信息,就能通过目录在它们之间建立一条证明路径。为此,每一个 CA 都必须存储一个证书和一个与其前面 CA 对应的反向证书。

7.6 证书包含在目录项中,其属性类型为 UserCertificate,CACertificate,和 CrossCertificatePair,目录

可以识别这些属性类型。使用其他属性一样的协议对这些属性类型进行操作。在本标准的3.3中给出了这些属性类型的定义。这些属性类型的规范如下：

```
UserCertificate      ::=ATTRIBUTE
    WITH ATTRIBUTE—SYNTAX Certificate
CACertificate        ::=ATTRIBUTE
    WITH ATTRIBUTE—SYNTAX Certificate
CrossCertificatePair ::=ATTRIBUTE
    WITH ATTRIBUTE—SYNTAX CertificatePair
CertificatePair      ::=
SEQUENCE{
        forward[0]Certificate OPTIONAL
        reverse[1]Certificate OPTIONAL
        ——至少应出现一个——}
```

用户可以从一个或多个证明职能机构得到一个或多个证书。并且每个证书都携带有发出该证书的证明职能机构的名字。

证书和证明路径可由以下ASN.1数据类型表示：

```
Certificates   ::=SEQUENCE{
    userCertificate          Certificate
    certificationPath        ForwardCertificationPath OPTIONAL}
CertificationPath  ::=SEQUENCE{
    userCertificate          certificate
theCACertificates SEQUENCE OF
                 CertificatePair OPTIONAL}
```

另外，正向证明路径可由以下ASN.1数据类型表示。这个成分包含能指回源发者的证明路径：

```
ForwardCertificationPath::=SEQUENCE OF CrossCertificates
```

7.7 在通常情况下，在用户能够相互鉴别之前，目录应必须提供这个完整的证书，并返回证明路径。但在实际操作过程中，对某个特定的鉴别实例来讲，通过如下途径可以减少从目录中获得的信息量。

a) 如果两个想要鉴别的用户具有同一个证明职能机构，那么，证明路径将变得毫无价值，而且用户可以相互直接打开彼此的证书；

b) 如果用户的CA是按层次安排的，那么一个用户可以存储用户与DIT根之间的所有证明职能机构的公开密钥、证书和反向证书。作为一种典型的情况，应包含只知道三个或四个证明职能机构的公开密钥和证书的用户。这样的用户只要求获得到公共信任点的证明路径。

c) 如果一个用户与被某个特定的其他CA证明的用户频繁的通信，则该用户只须从目录中获得其他用户的证书，从而取得从本地到那个CA的证明路径，并从这个CA返回这条证明路径。

d) 证明职能机构可以通过双方协商彼此进行交叉证明，从而可以缩短证明路径。

e) 如果两个用户以前曾经相互通信过，并且彼此已取得对方的证书，则它们无需援引目录资源就能相互鉴别。

不论是哪一种情况，用户从证明路径中取得其他每个用户的证书后，应检查收到的证书的有效性。

7.8 图4给出了假设的DIT段的示例，这里，CA为一层次结构。除了知道CA的信息外，还假定每一个用户都知道其证明职能机构的公开密钥，以及他自己的公开和秘密密钥。

7.8.1 如果用户的CA是按层次结构安排的，则A可以从目录中得到下列证书，以建立到B的证明路径：

X《W》，W《V》，V《Y》，Y《Z》，Z《B》

当 A 已经得到这些证书时，则可以按次序打开这个证明路径，进而得出 A(包括 A_P)的证书的内容：

$B_P=X_P \cdot X\langle\langle W\rangle\rangle W\langle\langle V\rangle\rangle V\langle\langle Y\rangle\rangle Y\langle\langle Z\rangle\rangle Z\langle\langle B\rangle\rangle$

一般情况下，A 还应从目录中得到下列证书，以建立从 B 到 A 的反向证明路径：

$Z\langle\langle Y\rangle\rangle, Y\langle\langle V\rangle\rangle, V\langle\langle W\rangle\rangle, W\langle\langle X\rangle\rangle, X\langle\langle A\rangle\rangle$

当 B 从 A 收到这些证书时，则可以按次序打开这个反向证明路径，进而得出 B(包括 B_P)的证书的内容：

$A_P=Z_P \cdot Z\langle\langle Y\rangle\rangle Y\langle\langle V\rangle\rangle V\langle\langle W\rangle\rangle W\langle\langle X\rangle\rangle X\langle\langle A\rangle\rangle$

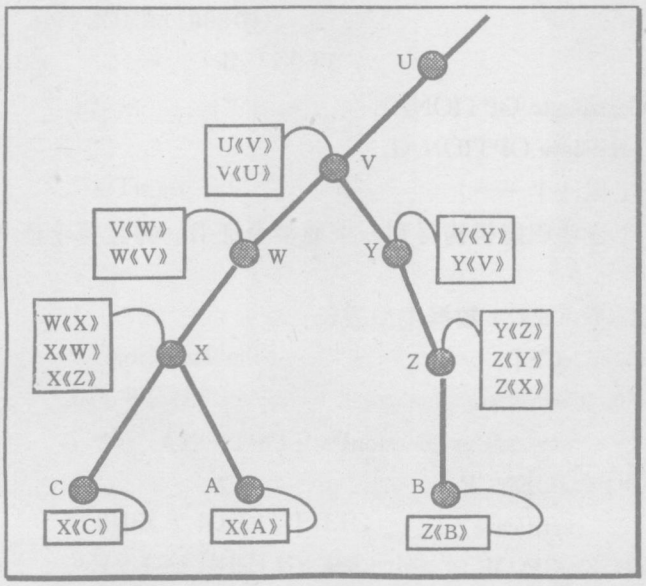

图 4 CA 的层次结构示例

7.8.2 可对 7.7 实施优化：

a) 例如，A 和 C 都知道 X 的公开密钥 X_P，因此，A 只须从目录中直接得到 C 的证书。需打开的证明路径可以缩减为：

$C_P=X_P \cdot X\langle\langle C\rangle\rangle$

并且，需打开的反向证明路径也可以缩减为：

$A_P=X_P \cdot X\langle\langle A\rangle\rangle$

b) 假定 A 知道 $W\langle\langle X\rangle\rangle$、$W_P$、$V\langle\langle W\rangle\rangle$、$V_P$、$U\langle\langle V\rangle\rangle$、$U_P$ 等等，则从目录中获取的用于构造证明路径的信息可以减少为：

$V\langle\langle Y\rangle\rangle$、$Y\langle\langle Z\rangle\rangle$、$Z\langle\langle B\rangle\rangle$

并且，从目录中获取的用于构造反向证明路径的信息为：

$Z\langle\langle Y\rangle\rangle$、$Y\langle\langle V\rangle\rangle$

c) 假定 A 与由 Z 证明的用户通信频繁，则他除了已知道上面 b)中的各个公开密钥外，还能知道 $V\langle\langle Y\rangle\rangle$，$Y\langle\langle V\rangle\rangle$，$Y\langle\langle Z\rangle\rangle$，和 $Z\langle\langle Y\rangle\rangle$。因此，为与 B 通信，A 只须从目录中获得 $Z\langle\langle B\rangle\rangle$即可。

d) 假定进行频繁通信的用户都由 X 和 Z 证明，那么在目录中，X 的目录项应持有 $X\langle\langle Z\rangle\rangle$，相反，Z 的目录项应持有 $Z\langle\langle X\rangle\rangle$(见图 4)。如果 A 想要鉴别 B，A 只需获得：

$X\langle\langle Z\rangle\rangle$、$Z\langle\langle B\rangle\rangle$

以构造证明路径；同样，只需获得：

$Z\langle\langle X\rangle\rangle$

即可构造反向证明路径。

e) 假定用户 A 和 C 以前曾经通信过，并且相互间已知彼此的证书，那么他们可以直接使用彼此的公开密钥；即：

$C_P = X_P \cdot X\langle\langle C\rangle\rangle$

$A_P = X_P \cdot X\langle\langle A\rangle\rangle$

7.8.3 在更一般的情况下，证明职能机构并不按层次结构相互联系。在图 5 中给出了一个假想的示例，在这个示例中，假定由 U 证明的用户 D 想要鉴别由 W 证明的用户 E。用户 D 的目录项持有证书 U《D》，用户 E 的目录持有证书 W《E》。

V 为一个 CA，证明职能机构 U 和 W 曾通过 V 按照可信任的途径相互交换过彼此的公开密钥。其操作结果是生成了证书 U《V》、V《U》、W《V》和 V《W》，并已存入目录中。假定 U《V》、W《V》存在 V 的目录项中，V《U》存在 U 的目录项中，而 V《W》存在 W 的目录项中。

用户 D 必须找到到用户 E 的证明路径。有几种方法可供使用。其中之一就是将用户和 CA 看作结点，而证书则为定向曲线图上的弧。在这里，D 应在曲线图上执行一次搜索以找到一条从 U 到 E 的路径，例如，U《V》、V《W》、W《E》。当找到这条路径以后，其反向路径亦可据此构造出来，即 W《V》、V《U》、U《D》。

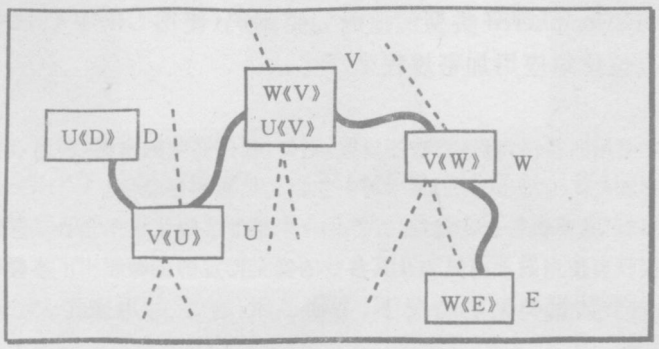

图 5 非层次结构的证明路径示例

8 数字签名

本章并不为数字签名规定一种通用的标准，但要规定在目录中用于签名权标的方法。

8.1 信息通常是将信息加密的摘要附加在该信息的后面来实现签名。信息的摘要则用一个单向散列函数产生，而加密则是用签名者的秘密密钥来执行的（见图 6），即：

$X\{Info\} = Info, X_S[h(Info)]$

注：使用秘密密钥进行加密可以保证签名不被伪造。而散列函数的单向特性则可保证不能从虚假的信息中产生出与正确结果相同的散列结果（及签名）。

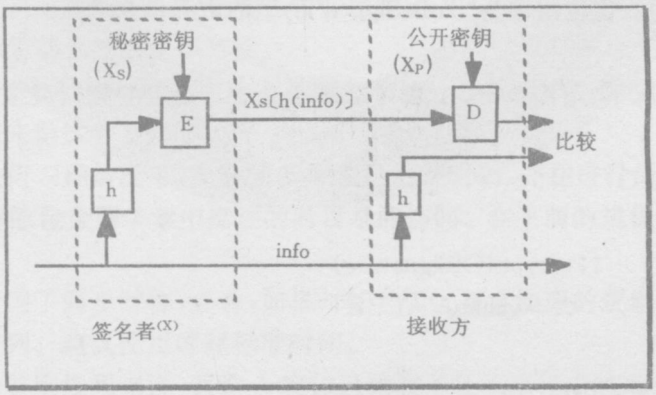

图 6 数字签名