ICS 35.100.70
L 79

中华人民共和国国家标准

GB/T 19714—2005

信息技术　安全技术　公钥基础设施　证书管理协议

Information technology—Security technology—Internet public key infrastructure—Certificate management protocol

2005-04-19 发布　　2005-10-01 实施

中华人民共和国国家质量监督检验检疫总局
中国国家标准化管理委员会　发布

国家图书馆专用

目　　次

国家图书馆专用

国家图书馆专用

前　言

本标准是依据 IETF RFC 2510 制定的。

本标准凡涉及密码算法相关内容，按国家有关法规实施。

本标准中引用的 RSA、SHA1、DH 密码算法均为举例性说明，具体使用时均须采用国家商用密码管理委员会批准的相应算法。

本标准的附录 B、附录 C、附录 F 为规范性附录，附录 A、附录 D、附录 E、附录 G 为资料性附录。

本标准由中华人民共和国信息产业部提出。

本标准由全国信息安全标准化技术委员会(TC260)归口。

本标准主要起草单位：北京创原天地科技有限公司、中国电子技术标准化研究所。

本标准主要起草人：林雪焰、吴志刚、王炳艳、陈震琦、张科研、李 丹、罗锋盈、陈 星。

国家图书馆专用

引　言

本标准描述了公钥基础设施(PKI)证书管理协议，定义了与证书产生和管理相关的各方面所需要的协议消息，主要包括：申请证书、撤销证书、密钥更新、密钥恢复、交叉认证等等。

公钥基础设施中总共有四类实体：CA、RA、终端实体、证书/CRL 库，如何保证四实体之间的通信安全、在证书业务中如何对四类实体进行管理，这些问题是本标准解决的主要问题。

国家图书馆专用

信息技术　安全技术　公钥基础设施
证书管理协议

1　范围

本标准描述了公钥基础设施(PKI)中的证书管理协议,定义了与证书产生和管理相关的各方面所需要的协议消息,这些消息主要包括申请证书、撤销证书、密钥更新、密钥恢复、交叉认证等。

本标准主要适用于在安全或不安全环境中实施 PKI 组件并实施管理,可作为 PKI 运营机构、PKI 组件开发者的参考指南。

2　规范性引用文件

下列文件中的条款通过本标准的引用而成为本标准的条款。凡是注日期的引用文件,其随后所有的修改单(不包括勘误的内容)或修订版均不适用于本标准,然而,鼓励根据本标准达成协议的各方研究是否可使用这些文件的最新版本。凡是不注日期的引用文件,其最新版本适用于本标准。

GB/T 16264.8—2005　信息技术　开放系统互连　目录　第8部分:公钥和属性证书框架(ISO/IEC 9594-8:2001,IDT)

RFC2511 因特网 X.509 公开密钥基础设施证书消息格式

3　术语和定义

下列术语和定义适用于本标准。

3.1

抽象语法记法一(ASN.1)　Abstract Syntax Notation 1(ASN.1)

用来组织复杂数据对象的表示法。

3.2

公钥证书　public key certificate

用户的公钥连同其他信息,并由发布该证书的证书认证机构的私钥进行加密使其不可伪造。

3.3

证书持有者　certificate holder

有效证书的主体对应的实体。

3.4

证书用户　certificate user

需要确切地知道另一实体的公开密钥的某一实体。

3.5

证书认证机构(CA)　Certificate Authority(CA)

负责创建和分配证书,受用户信任的权威机构。用户可以选择该机构为其创建密钥。

3.6

证书认证路径　certification path

一个 DIT 中对象证书的有序序列,通过处理该有序序列及其起始对象的公钥可以获得该路径的末端对象的公钥。

国家图书馆专用

3.7

认证业务说明(CPS)　Certification Practice Statement(CPS)

证书认证机构发放证书时遵循的业务说明。

3.8

交叉证书　cross-certificate

两个 CA 间为交叉认证所互相签发的数字证书。

3.9

CRL 分布点　CRL distribution point

一个 CRL 目录项或其他 CRL 分发源;由 CRL 分布点分发的 CRL 可以包括仅对某 CA 所发证书全集某个子集的撤销条目,或者可以包括有多个 CA 的撤销条目。

3.10

证书撤销列表(CRL)　Certificate Revocation List(CRL)

一个已标识的列表,它指定了一套证书发布者认为无效的证书。除了普通 CRL 外,还定义了一些特殊的 CRL 类型用于覆盖特殊领域的 CRL。

3.11

发证　certify

颁发一个证书的行为。

3.12

可辨别编码规则(DER)　Distinguished Encoding Rules(DER)

对 ASN.1 对象进行编码的规则。

注:本标准中使用 DER 对 ASN.1 对象进行编码。

3.13

数字签名　digital signature

允许接收者验证签名人的身份和数据完整性的数据单元。

3.14

目录服务(DS)　Directory Service(DS)

分布在网络中的各种节点或服务器提供的分布式数据库服务。

3.15

终端实体　end entity

不以签署证书为目的而使用其私钥的证书主体或者是依赖(证书)方。

3.16

散列函数　hash function

哈希函数

将值从一个大的(可能很大)定义域映射到一个较小值域的(数学)函数。“好的”散列函数是把该函数应用到大的定义域中的若干值的(大)集合的结果可以均匀地(和随机地)被分布在该范围上。

3.17

散列码　Hash code

散列函数的输出比特串。

3.18

消息认证码(MAC)　Message Authentication Code(MAC)

通过密码技术由消息产生的认证数据。

国家图书馆专用

3.19

消息摘要　message digest

散列一个消息后得到的固定长度数据。

3.20

个人安全环境(PSE)　Personal Security Environment (PSE)

证书及私钥的终端实体的安全本地存储。

3.21

拥有证明(POP)　Proof of Possession (POP)

终端实体用以证明自己拥有(即能够使用)与为之申请证书的公钥相对应的私钥。

3.22

策略映射　policy mapping

当某个域中的一个 CA 认证另一个域中的一个 CA 时,在第二个域中的特定证书策略可能被第一个域中的证书认证机构认为等价(但不必在各方面均相同)于第一个域中认可的特定证书策略。

3.23

注册机构(RA)　Registration Authority(RA)

为用户办理证书申请、身份审核、证书下载、证书更新、证书注销以及密钥恢复等实际业务的办事机构或业务受理点。

3.24

资料库　repository

存储证书和 CRL 等信息,并提供无需验证的信息检索服务的数据库。

3.25

自颁发证书　self-issued certificate

证书的主体和颁发者相同的 CA 证书。

国家图书馆专用

4　缩略语

下列缩略语适用于本标准:

CA　证书认证机构

CRL　证书撤销列表

PKCS　公钥密码系统

PKI　公钥基础设施

POP　拥有证明

PSE　个人安全环境

RA　注册机构

TCP　传输控制协议

5　PKI 管理概述

5.1　PKI 管理模型

在详细阐述特定的消息格式和流程之前,首先要定义一下 PKI 管理中所涉及到的实体以及它们之间的交互行为(根据 PKI 管理所需的功能)。然后再对这些功能进行归类,以包含可以确定的不同类型的终端实体。

5.2　PKI 实体的定义

PKI 管理中所涉及到的实体包括终端实体和证书认证机构。注册机构也可以被包括在内。

5.2.1 主体和终端实体

终端实体是不以签署证书为目的而使用其私钥的证书主体或者是依赖(证书)方。

“主体”在此处是指被授予证书的实体,一般在证书的主体或主体可替换名字段中指定。当要区分主体所使用的工具和/或软件时(例如:一个本地的证书管理模块),使用术语“主体设施”。通常优先使用术语“终端实体”而不是“主体”,以防止与证书字段名中的主体相混淆。

需要着重指出的是:终端实体在此处不仅包括应用软件的使用者,也包括软件本身(例如 IPSec 的情况)。这一因素影响着 PKI 管理操作所使用的协议。例如,与个人用户相比,应用软件更有可能确切知道需要使用证书中的哪个扩展项。PKI 管理实体也被看作为一个终端实体,因为它们有时候也在证书(或交叉证书)的主体或主体可替换名字段中被指定。如无特殊说明,术语“终端实体”将不会被用来指代 PKI 管理实体。

所有的终端实体都需要安全可靠的访问一些本地信息,至少包括:实体自己的名字和私钥,被该实体直接信任的 CA 的名字及其公钥(或公钥指纹,如果可以通过其他的方式得到自签名证书)。软件实现可以使用安全本地存储机制存储上述信息,但不限于上述信息(例如:还可以包括用户自己的证书以及应用软件特有的信息)。存储方式也可以不同,例如普通的文件或抗攻击的密文存储介质。像这样安全的本地存储在这里被称之为终端实体的个人安全环境。

尽管有关 PSE 格式的内容已经超出本标准的范围(与设备及其他信息相关),但这里定义了一种通用的 PSE 数据交换格式 —— 认证响应消息。

5.2.2 证书认证机构(CA)

证书认证机构是负责创建和分配证书,受用户信任的权威机构。用户可以选择该机构为其创建密钥。

从终端实体的角度出发,CA 可以是,也可以不是一个事实上真正的“第三方”。绝大多数情况下,CA 与其所服务的终端实体属于同一个机构。

同样,我们使用术语“CA”指代证书中签发者(issuer)字段所代表的实体。当需要与 CA 所使用的软硬件工具相区分时,我们使用术语“CA 设施”。

CA 设施一般包括离线模块和在线模块,只有 CA 的离线模块可以使用 CA 的私钥。尽管是否这样做与 CA 的策略相关,但这取决于软件实现者。

我们使用术语“根 CA”来指代被终端实体直接信任的 CA;即安全地获取根 CA 的公钥需要一些额外的步骤。这一术语并不意味着根 CA 必须处于 CA 体系层次的顶层,它只是说明该 CA 是被终端实体直接信任的。

“下级 CA”(子 CA)不是终端实体的根 CA。通常,下级 CA 不是任何实体的根 CA,但这并不是强制的。

5.2.3 注册机构(RA)

注册机构是为用户办理证书申请、身份审核、证书下载、证书更新、证书注销以及密钥恢复等实际业务的办事机构业务受理点,亦称证书注册审核中心。

除了终端实体和 CA 之外,很多应用环境要求把注册机构从证书认证机构中独立出来。(RA 存在的原因参见附录 A。)RA 所具有的功能将因情况不同而有所不同,可能包括个人身份鉴别、介质分发、作废报告、名称分配、密钥产生、密钥归档等等。

本标准将 RA 作为可选的组成部分,当 RA 不存在时,假定 CA 可以实现 RA 的功能。从终端实体的观点看,它们都使用相同的 PKI 管理协议。

同样,当需要区分 RA 和 RA 所使用的工具时,使用“RA 设施”这个术语。

应该注意到 RA 本身也是一个终端实体,一个被验证了的终端实体,它有自己的私钥进行数字签名和身份认证。CA 设施如何把某些终端实体认定为 RA 则是一个实现问题(本标准不阐述特定的 RA 认证操作)。并不强制 RA 必须可以被与其正通信的 CA 所认证(所以一个 RA 可以只被认证一次,与多

国家图书馆专用

个 CA 协同工作)。

在某种情况下,即使存在 RA,终端实体可能仍然需要和 CA 直接通信。例如,在 RA 进行登记和接受审核,而直接与 CA 通信来更新证书。

5.3 PKI 管理要求

PKI 管理的要求如下:

a) PKI 管理必须符合 GB/T 16264.8—2005 及相关修订补篇(指证书扩展项);

b) PKI 管理必须符合 PKI 系列草案的其他标准;

c) 必须能够在不影响其他密钥对的前提下定期地更新任意密钥对;

d) 为了使调整简单易行,在 PKI 管理协议中应尽可能少的使用加密;

e) PKI 管理协议必须允许使用不同的工业标准加密算法。这意味着,原则上,任何给定的 CA、RA 或终端实体,可以使用任何适合其所拥有的密钥对的算法;

f) PKI 管理协议一定不能排除密钥对由相关的终端实体或 RA 或 CA 产生。密钥产生也可以在其他地方完成,出于 PKI 管理的目的,我们可以认为密钥生成发生在密钥第一次出现的地方:终端实体、RA 或 CA;

g) PKI 管理协议必须能够支持相关终端实体或 RA、CA 进行证书发布。在不同实现和不同的环境下可以采用上面任何一种方法;

h) PKI 管理协议必须允许通过认证的终端实体请求作废证书,以签发 CRL。这一功能必须能够尽可能防止拒绝服务攻击;

i) PKI 管理协议必须能够使用各种不同的传输机制,特别是邮件传输协议、HTTP、TCP/IP 和 FTP;

j) 签发证书的最终决定权属于 CA。RA 或终端实体都不能假定 CA 签发出来的证书符合它们的全部要求。CA 可以根据其运营策略,改变一个证书字段的值,或者添加、删除、修改证书扩展项。换句话说,所有的 PKI 实体(终端实体、RA、CA)都必须能够处理那些与其请求不一致的证书响应(例如:CA 可能会缩短证书有效期)。CA 策略可能作出规定,在证书请求者检查并接受新签发的证书之前,CA 机构不能发布或分发该证书(通常使用证书确认消息完成这一功能);

k) PKI 管理协议必须能够支持未泄密的 CA 密钥对的更新(即 CA 密钥更新)。如果 CA 发生密钥泄露,那么该 CA 所辖的全部实体都必须重新进行初始化操作。在 CA 密钥更新以后,PSE 中包含 CA 新公钥的终端实体,必须仍然能够验证使用旧的 CA 公钥能够验证的证书。直接信任旧的 CA 密钥对的终端实体,也必须能够验证新的 CA 私钥所签发的证书;

l) 在某些实现或情况下,RA 的功能可能会由 CA 来承担。PKI 管理协议的设计,必须满足下面的条件:终端实体不管与 CA 还是与 RA 通信都应该使用相同的协议;

m) 当终端实体发出的证书请求带有公钥时,必须能够证明拥有相应的私钥。完成此项工作可以用不同方法,究竟采用哪一种取决于认证请求的类型。关于证书管理协议消息中定义的带内方法完成以上工作的详细内容见 6.3。

5.4 PKI 管理操作

图 1 描述了 PKI 管理操作所定义的各个实体间的关系。可以沿着字母标明的线路发送 PKI 消息。

国家图书馆专用

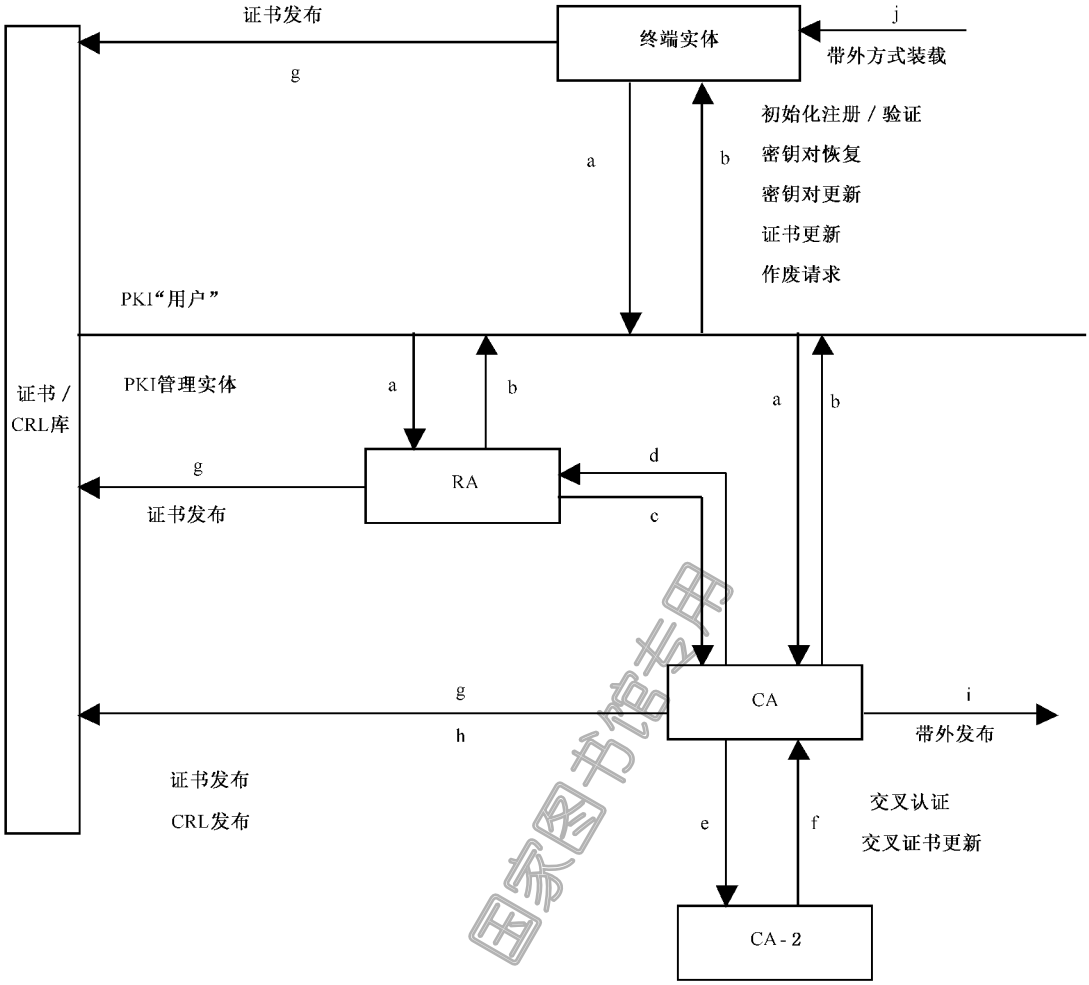

图 1　PKI 实体

在上层，各种 PKI 操作可以按下列方式分组：

a）　CA 的建立。当建立一个新的 CA 时，需要完成一些必要的操作（例如：发布初始 CRL，导出 CA 公钥）。

b）　终端实体初始化：包括导入根 CA 公钥，以及获得 PKI 管理实体所支持的可选项信息。

c）　认证：各种操作都将导致创建新的证书：

1）　初始注册和认证：终端实体在 CA 为它签发证书之前第一次向 CA 或 RA 表明自己的身份。这一过程成功时的最终结果是 CA 为终端实体的公钥签发证书，并将该证书返回给终端实体和/或将该证书发布到公共的数据仓库中。这一过程通常包括多个“步骤”，还可能包括终端实体设施的初始化过程。例如，终端实体设施必须能够安全地导入 CA 公钥，以用来验证证书路径。此外，终端实体通常还需要导入自己的密钥对。

2）　密钥对更新：任何密钥对都需要定期更新（即用新的密钥对替换旧的密钥对），因此需要签发一个新的证书。

3）　证书更新：由于证书会过期，如果其他相关信息没有变化，那么证书就可以被“刷新”。

4）　CA 密钥对更新：同终端实体一样，CA 密钥对也需要定期更新，但需要不同的机制。

国家图书馆专用

5) 交叉认证请求：一个 CA 请求另一个 CA 签发一个交叉证书。“交叉证书”的主体 CA 与签发者 CA 完全不同，主体公钥信息(SubjectPublicKeyInfo)字段包含着验证密钥(即该证书是为主体 CA 的签名密钥对签发的)。交叉证书细致的区分时使用下面的术语：“域间交叉证书” —— 如果交叉证书的主体 CA 和签发者 CA 属于不同的管理域；其他的交叉证书则称作“域内交叉证书”。

注 1：上面所定义的“交叉证书”与 X.509 中定义的“CA 证书”是并列的。注意不要把“交叉证书”同 X.509 中的“cACertificate”属性相混淆，它们之间没有联系。

注 2：除非特别说明，否则一般认为术语“交叉证书”指的就是“域间交叉证书”。

注 3：交叉证书的签发可以是相互的(但并非必须这样做)；换句话说，两个 CA 可能彼此为对方签发交叉证书。

6) 交叉证书更新：与普通证书更新类似，不同之处就在于它操作的是交叉证书。

d) 证书/CRL 搜索操作：某些 PKI 管理操作将导致发布证书或 CRL。

1) 证书发布：产生证书之后，就需要通过一些方法发布这些证书。PKIX 中定义的方法可以是 7.3.13～7.3.16 中规定的方法，或者是 RFC2559、RFC2585(PKIX 系列规范的“操作协议”方案)中描述的其他方法(例如：LDAP)。

2) CRL 发布：与证书发布类似。

e) 恢复操作：当一个 PKI 实体“丢失”它的 PSE 时，需要通过某些 PKI 操作完成恢复工作。

密钥对恢复：作为可选操作，用户密钥资料(例如用户的解密密钥)可以由 CA、RA 或与 CA 或 RA 相关的密钥备份系统进行备份。如果一个实体需要恢复它已经备份的密钥资料(例如忘记了私钥保护密码或丢失了密钥链文件)，就需要一个支持密钥恢复的协议消息。

f) 撤销操作：某些 PKI 操作将需要创建新的 CRL 条目或新的 CRL。

撤销请求：一个经过授权的人向 CA 发出异常情况警告并要求撤销证书。

g) PSE 操作：PSE 操作(例如转移 PSE、更改口令等)的定义超出了本标准的范围，这里我们还是定义了一个 PKI 消息(CertRepMessage)，它可以作为该操作的基础。

注：在线协议并非是实现以上操作的唯一途径。对任何一个操作来说，离线方法可以达到同样的效果，本标准也并不强制使用在线协议。例如：当使用硬件介质时，很多操作都通过物理介质的传送来完成。

后面将定义一组支持以上操作的标准消息。关于在不同环境中传送这些消息的传输协议(基于文件的、在线、E-mail 和 WWW)的定义，已经超出了本标准的范围，将在其他文档中单独说明。

6 前提与限制

6.1 终端实体初始化

对于每一个终端实体，在与 PKI 管理实体进行交互之前，首先要申请获得一些信息，包括获得 PKI 系统所支持的功能、安全获得相关根 CA 的公钥。

6.2 初始注册/认证

终端实体的初始化注册和认证有很多种方案。由于 CA 执行的策略各不相同，并且终端实体的类型也有所不同，因此没有一种方案能适应所有的环境。

初始化注册前，终端实体与 PKI 还没有任何的联系。当终端实体已经拥有认证过的密钥对时，就可以进行简化或选择其他的方案。

以下对本标准支持的初始化注册和认证方案进行分类。方案中一部分为强制性的，一部分为可选择的。强制性的方案能够覆盖到大多数的实际应用，而可选方案满足那些不是很常用的应用。这样，在系统的灵活性和系统实现的简便上进行了折衷。

6.2.1 所用的准则

6.2.1.1 注册/认证的启动

就产生的 PKI 消息而言，初始化注册/认证的启动发生在产生第一条与终端实体相关的 PKI 消息

的地点。可能的场所是在终端实体，RA 或者 CA。

6.2.1.2 终端实体消息的最初认证

终端实体产生的请求证书的在线消息认证与否都可以，但要求对最初终端实体发给 PKI(CA/RA)的任何消息进行认证。

在本标准中，PKI(CA/RA)通过带外方式向终端实体发放秘密数据(初始认证密钥)和参考数据(用于识别密钥)来完成。然后初始认证密钥就可以用来保护相关的 PKI 消息了。

因此，通过判断在线消息是否经过初始鉴别从而对初始注册/认证方案进行分类。

注 1：本标准不讨论对 PKI 系统发给终端实体的消息的认证，因为在所有情况下，在终端实体设施安装根 CA 的公钥或基于初始认证密钥都可以实现这一认证。

注 2：如果终端实体发送的消息通过带外方式被鉴别，那么可以认为初始注册/认证流程是安全的。

6.2.1.3 密钥产生的地点

在本标准中，认为“密钥产生”的地点是密钥(无论是公钥还是私钥)第一次在 PKI 消息中出现的地方。注意，这不排除有一个密钥集中产生的服务，密钥可以在其他地方产生，然后使用一个(自定义的或者标准的)密钥产生请求/响应协议(该协议不在本标准的讨论范围之内)导入到终端实体、RA 或 CA 中。

密钥产生的地点有三种可能：终端实体、RA 或者 CA。

6.2.1.4 成功认证的确认

在为一个终端实体创建初始证书以后，通过让终端实体显示确认成功地接收到了包含证书(或表明证书已生成)的消息，从而可以得到附加的保证。当然，这个确认消息必须受到保护(通过初始认证密钥或者其他方式)。

这又提供了两种可能性：确认的或者未确认的。

6.2.2 强制的方案

上面的标准考虑到了大多数的初始注册/认证方案。要求符合本标准的 CA 设施、RA 设施和终端实体设施必须支持下面的第二种方案。如果需要，任何实体还可以支持其他的方案。

6.2.2.1 集中方案

按照上面的分类标准，这种方案可能是最简单的：

—— 启动发生在进行认证的 CA；

—— 不要求在线消息的认证；

—— 密钥由进行认证的 CA 产生(见 6.2.1.3)；

—— 不要求确认消息。

从消息流的角度来看，本方案意味着只要求一个由 CA 发送给终端实体的消息。这个消息必须包含终端实体的全部 PSE 信息。该消息使用加密传输，通过带外方式通知终端实体(如用电话通知消息的保护口令)，使终端实体认证并解密接收到的消息。

6.2.2.2 基本认证方案

按照上面的分类标准，本方案如下：

—— 启动发生在终端实体；

—— 要求进行消息认证；

—— 密钥由终端实体产生(见 6.2.1.3)；

—— 要求确认消息。

从消息流的角度来看，基本认证方案见图 2：

国家图书馆专用

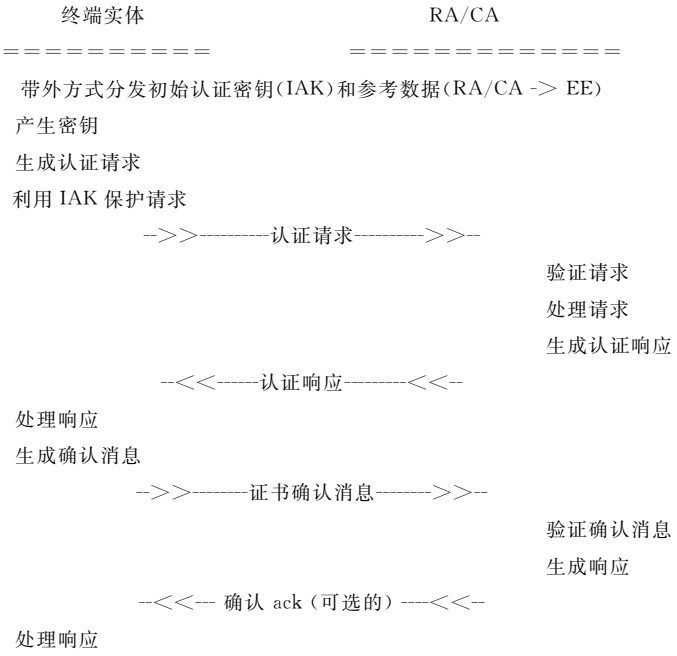

图 2　初始化注册/认证基本方案

在验证证书确认消息失败的情况下，如果新签发的证书已经发布或者可以由其他的方式获得，那么CA/RA 必须撤销该证书。

6.3　私钥拥有证明

为了使 CA 和 RA 能够有效地验证终端实体和密钥对间的绑定是否合法，这里定义一种 PKI 管理操作，终端实体可以通过这种操作证明自己拥有并能使用与申请证书的公钥相对应的私钥。一个 CA/RA 在认证交换时可以自由选择如何实现 POP(例如带外方式或带内方式)。目前有很多非 PKIX 的操作协议(例如各种电子邮件协议)，并不明确验证终端实体和私钥间的绑定；因此要求 CA/RA 必须通过某种方式实现 POP。在验证绑定(签名、加密和密钥协商密钥对)的操作协议广泛存在之前，绑定的验证只能被 RA/CA 实现。因此，没有被 RA/CA 验证绑定的 PKI 证书就会因为没有任何意义而终止。

按照证书申请中的密钥对类型的不同，POP 须通过不同的方式完成。如果是一个多种用途的密钥，那么，任意一种适当的方法都可以使用(例如，如果一个密钥可以用于签名以及其他应用，那么就不能将它送到 CA/RA 去验证 POP)。

本标准允许终端实体向 RA 提供相关证明，随后 RA 向 CA 说明需要的证明已经完成(并验证)。例如：一个想要申请签名证书的终端实体向 RA 送去签名信息，RA 随后通知 CA 终端实体已经提供了要求的证明。当然，某些策略可能不允许这种情形(例如，CA 可能是认证过程中唯一可以验证 POP 的实体)。

6.3.1　签名密钥

对于签名密钥的情况，终端实体通过对一个数据签名来证明拥有私钥。

6.3.2　加密密钥

对于加密密钥的情况，终端实体可以将私钥提供给 CA/RA，或者通过解密一段数据来证明对私钥的拥有(见 7.2.8)。解密数据可以是直接方式也可以是间接方式。

直接方式是 RA/CA 给终端实体发送一个随机的挑战数，同时要求终端实体立即返回响应。

间接方式是给终端实体返回加密的证书(让终端实体在确认消息中证明它能解开这个证书)。这使得只有预期的终端实体才能使用 CA 签发的证书。

本标准鼓励使用间接方式，因为这种方式不需要发送额外的消息（即可以使用消息三元组{请求，响应，确认}来证明)。

6.3.3 协商密钥

对于协商密钥的情况，为了证明终端实体拥有私钥，终端实体和 PKI 实体（RA 或者 CA）必须建立一个共享的密钥。

注意，这种方法并不需要对 CA 可以认证的密钥做什么限制 —— 对于 Diffie-Hellman 密钥，终端实体可以自由选择它的算法参数 —— 只要 CA 在需要时可以利用适当的参数生成短期（或一次性）的密钥。

6.4 根 CA 的更新

这里只描述对某些终端实体来说是根 CA 的情况。

这里所描述流程的基础是 CA 利用旧私钥保护新私钥同时利用新私钥保护旧私钥。因此，当 CA 更新密钥对时，如果证书要在 X.500 目录服务器发布，则必须生成两个额外的 cACertificate 属性值（这样共有四个：oldWithOld、oldWithNew、newWithOld 以及 newWithNew）。

当一个 CA 更改密钥对的时候，那些已经通过带外方式持有该 CA 旧公钥的终端实体受到的影响最大。这些终端实体需要访问由旧私钥保护的 CA 新公钥。然而，它们只在一个有限的时间段需要这一证书（直到它们通过带外方式获得新的 CA 公钥）。通常，当这些终端实体的证书到期的时候，这自然便实现了。

用来保护新旧 CA 公钥的数据结构不是新的数据结构，是标准的证书结构（也可能包含扩展项）。

注 1：为了支持 v1 版本证书，目前的方案没有利用任何 X.509v3 的证书扩展项。密钥标识（KeyIdentifier）扩展项的存在是为了提高效率。

注 2：可以将这一方案进行推广，以适应 CA 在它的任一终端实体证书的有效期内多次更新自己密钥对的情形，但这种推广似乎没有什么价值。不作这种推广仅仅意味着 CA 密钥对的有效期必须比用这对密钥签发的所有证书的有效期都长。

注 3：本方案确保终端实体最晚将在它所拥有的由 CA 旧公钥签发的最后一个证书过期时获得 CA 新公钥（通过带外方法）。发生在其他时候的证书和/或密钥更新操作并不要求这一点（这依赖于终端实体设施）。

6.4.1 CA 操作员的行为

要更新 CA 的密钥，CA 操作员要完成以下操作：

a） 产生新密钥对；

b） 产生一个用新私钥为旧公钥签名的证书（“old with new”证书）；

c） 产生一个用旧私钥为新公钥签名的证书（“new with old”证书）；

d） 产生一个用新私钥为新公钥签名的证书（“new with new”证书）；

e） 通过数据仓库或其他方式发布这些新证书（可能使用 CAKeyUpdAnn 消息）；

f） 导出 CA 新公钥，这样终端实体就可以通过带外方式获得（如果需要的话）。

旧 CA 私钥将不再使用。但旧 CA 公钥还会延续使用一段时间。当 CA 所属的终端实体都已经安全地获得了 CA 新公钥的时候，旧公钥就不再使用了（防抵赖除外）。

“old with new”证书的有效期必须从旧密钥生成的时间开始，到旧公钥到期的时候结束。

“new with old”证书的有效期必须从新密钥对生成的时间开始，到足以保证所有终端实体都得到了新的 CA 公钥时结束（最晚到旧公钥到期的时候）。

“new with new”证书的有效期必须从新密钥对生成的时间开始，到 CA 下一次更新其密钥时或 CA 下一次更新其密钥之前结束。

6.4.2 验证证书

通常在验证签名的时候，验证者要验证包含签名者公钥的证书。然而，一旦允许 CA 更新密钥，就会出现很多新的可能性，如表 1 所示：

国家图书馆告样

表 1　验证签名可能情况

	密钥库包含了新的和旧的公钥		密钥库只包含旧公钥(例:推迟发布)	
	PSE 包含新公钥	PSE 包含旧公钥	PSE 包含新公钥	PSE 包含旧公钥
签名者的证书由新公钥来保护	情况 1:这是一种标准的情况,验证者不需要访问数据仓库,可以直接验证证书	情况 3:这种情况下验证者必须访问数据仓库以获得新公钥	情况 5:CA 操作员没有更新数据仓库,但验证者可以直接验证证书(与第一种情况相同)	情况 7:在这样的情况之下 CA 操作员没有更新数据仓库,因此验证过程将会失败
签名者的证书由旧公钥来保护	情况 2:在这样的情况下,验证者要提必须访问数据仓库获取旧公钥	情况 4:在这样的情况下,验证者不需要访问数据仓库,可以直接验证	情况 6:验证者认为这种情况和情况 2 是一样的,会访问数据仓库,但是验证会失败	情况 8:CA 操作员没有更新数据仓库,但验证者可以直接验证,与情况 4 相同

6.4.2.1　在情况 1、情况 4、情况 5 和情况 8 下的验证

在这样的情况下,验证者拥有 CA 公钥的一份本地拷贝,可以用来直接验证证书。这种情况与没有密钥更新时是一样的。

第 8 种情况会在 CA 操作员已经生成了新密钥和将新密钥导入系统之间的时候出现。当此间隙期间 CA 操作员已经发布了签名者和验证者的证书的时候,情况 5 会出现(CA 操作员要尽量避开这样的情况)。

6.4.2.2　在情况 2 下的验证

在情况 2 下,验证者必须得到 CA 的旧公钥。验证者执行下列操作:

a)　在数据仓库中查找 caCertificate 属性,获得 OldWithNew 证书(基于有效期来判断,注意主体和签发者字段必须匹配);

b)　利用 CA 新密钥(验证者本地拥有的)验证该证书是否正确;

c)　如果正确,利用 CA 旧密钥验证签名者的证书。

当 CA 操作员先为签名者签发了证书,更新 CA 密钥后,又为验证者签发证书时,会发生这种情况,因此这是一种很典型的情况。

6.4.2.3　在情况 3 的验证

在第 3 种情况下,验证者必须要取得 CA 的新公钥,验证者执行下列操作:

a)　在数据仓库中查找 caCertificate 属性,获得 NewWithOld 证书(基于有效期来判断,注意主体和签发者字段必须匹配);

b)　利用 CA 旧密钥(验证者本地拥有的)验证该证书是否正确;

c)　如果正确,利用 CA 新密钥验证签名者的证书。

当 CA 操作员先为验证者签发了证书,更新 CA 密钥后,又为签名者签发证书时,会发生这种情况,因此这也是一种很典型的情况。

6.4.2.4　在情况 6 下的验证失败

在这种情况下 CA 已经为验证者签发了包含新密钥的 PSE 但尚未更新数据仓库中的属性。这意味着验证者无法得到可信赖的 CA 的旧密钥,所以验证失败。

此失败是由 CA 操作员的错误造成的。

6.4.2.5　在情况 7 下的验证失败

在这种情况下,CA 已经用新密钥为签名者签发了证书但尚未更新数据仓库中的属性。这意味着验证者无法得到可信赖的 CA 旧密钥,所以验证失败。

此失败也是由 CA 操作员的错误造成的。

6.4.3　作废 —— CA 密钥的改变

如上所述,一旦允许更改 CA 密钥,那么证书的验证将变得更为复杂。对于作废检查也存在同样的

国家图书馆专用

问题，因为 CA 可能利用新私钥而不是用户 PSE 中已经存在的私钥签发 CRL。

对各种情形的分析与证书验证相同。

7 数据结构

7.1 PKI 消息综述

本标准中所有用于 PKI 管理的消息都采用下面的结构：

```
PKIMessage ::= SEQUENCE {
        header              PKIHeader,
        body                PKIBody,
        protection [0]      PKIProtection OPTIONAL,
        extraCerts [1]      SEQUENCE SIZE (1..MAX) OF Certificate OPTIONAL
}
PKIMessages ::= SEQUENCE SIZE (1..MAX) OF PKIMessage
```

PKIHeader 包含许多 PKI 消息公有的信息。

PKIBody 包含与具体消息相关的信息。

PKIProtection 字段是可选的，包含对 PKI 消息进行保护的比特串。

extraCerts 字段可以包含对接收者来说可能有用的证书。例如，CA 或 RA 可以利用这一字段向终端实体提供它验证自己的新证书时所需要的证书(例如，如果签发终端实体证书的 CA 对它来说不是一个根 CA)。注意，这个字段不一定包含一个证书链 —— 为了使用证书，接收者可能需要对这些证书进行分类、选择或其他处理。

7.1.1 PKI 消息头

所有的 PKI 消息都要求使用消息头的某些信息进行寻址和交易标识。某些同样信息也出现在与传输相关的信封中。不管采用哪种方法，只要 PKI 消息受到保护，这些信息就会同样受到保护。

下面的数据结构用于保存这些信息：

```
PKIHeader ::= SEQUENCE {
        pvno                INTEGER { cmp1999(1), cmp2000(2) },
        sender              GeneralName,
        -- 标识发送者
        recipient           GeneralName,
        -- 标识预期的接收者
        messageTime     [0] GeneralizedTime         OPTIONAL,
        -- 产生这个消息的时间(当发送者认为该时间的传输合适时使用
        -- 即接收方接收到消息时这个时间仍有意义)
        protectionAlg   [1] AlgorithmIdentifier     OPTIONAL,
        -- 用于计算 protection 比特的算法
        senderKID       [2] KeyIdentifier           OPTIONAL,
        recipKID        [3] KeyIdentifier           OPTIONAL,
        -- 标识用于保护的特定密钥
        transactionID   [4] OCTET STRING            OPTIONAL,
        -- 标识交易；即在相关的请求、响应和确认消息中，该字段值相同
        senderNonce     [5] OCTET STRING            OPTIONAL,
        recipNonce      [6] OCTET STRING            OPTIONAL,
        -- 用于防止重放攻击的随机数，senderNonce 由消息的创建者赋值
```

国家图书馆

```
        -- recipNonce 是由本消息的预期接收者先前插入到相关消息中的随机数
        freeText        [7] PKIFreeText                 OPTIONAL,
        -- 可以用于表示上下文相关的说明(本字段主要由人工使用)
        generalInfo     [8] SEQUENCE SIZE (1..MAX) OF
                            InfoTypeAndValue        OPTIONAL
        -- 可以用于表示上下文相关的说明(本字段不是供人工使用的)
}
PKIFreeText ::= SEQUENCE SIZE (1..MAX) OF UTF8String
        -- 按 UTF-8[RFC2279]编码的文本(注意:每一个 UTF8String 都可以包含
        -- 一个 RFC 1766/RFC 3066 语言标签,用于表示所包含文本属于哪一种语言)
        -- 详情请参见 RFC2482
```

对于该版本 pvno 字段取固定值 2。

sender 字段包含 PKIMessage 发送者的名字。这个名字(与 senderKID 一起,如果提供的话)应当能够标识出对这一消息的保护进行验证所需要的密钥。如果发送方实体并不清楚自己的任何信息(例如:在初始化请求消息中,终端实体并不清楚自己的 DN、电子邮件、IP 地址等),那么"sender"字段必须包含一个"NULL"值;即一个长度为 0 的 SEQUENCE OF。在这种情况下,senderKID 字段必须包含一个标识符(即一个参考数),这个标识符能够向接收者表明验证消息所用的共享密钥信息。

recipient 字段包含 PKIMessage 接收者的名字。这个名字(与 recipKID 一起,如果提供的话)应当可以用于验证对消息的保护。

protectionAlg 字段指明保护消息所用的算法。如果没有提供保护比特位(注意 PKIProtection 是可选的),那么必须省略这个字段;如果提供了比特位,那么必须提供这个字段。

senderKID 和 recipKID 用于标识使用了哪个密钥保护消息(recipKID 通常仅在使用 Diffie-Hellman 密钥保护消息时才要求)。如果要求唯一标识一个密钥就必须使用这些字段(例如:如果一个发送者的名字与多个密钥相关联),否则就应当省略这些字段。

transactionID 字段用于使消息的接收者将这个消息与以前签发请求相关联。比如对于 RA,transactionID 可以将某消息与同一时刻众多消息区分开来。

senderNonce 和 recipNonce 字段保护 PKIMessage 免受重放攻击。

messageTime 字段包含发送者创建这个消息的时间。可以由终端实体用于校正/检查自己的本地时间以便与中央系统的时间保持一致。

freeText 字段可以用于发送一个人可读的消息给接收者(用任意多种语言)。这个序列中的第一种语言表明响应期望使用的语言。

generalInfo 字段可以用于向接收方发送机器可读的额外的数据。可以支持下面定义的 generalInfo 扩展项。

7.1.2 PKI 消息体

```
PKIBody ::= CHOICE { -- 与消息类型相关的消息体和参考章条
ir          [0]     CertReqMessages,            --初始化请求(7.3.1)
ip          [1]     CertRepMessage,             --初始化响应(7.3.2)
cr          [2]     CertReqMessages,            --认证请求(7.3.3)
cp          [3]     CertRepMessage,             --认证响应(7.3.4)
p10cr       [4]     CertificationRequest,       --PKCS #10 认证请求[PKCS10]
popdecc     [5]     POPODecKeyChallContent      --POP 挑战(7.2.8)
popdecr     [6]     POPODecKeyRespContent,      --POP 响应(7.2.8)
```

```
kur         [7]   CertReqMessages,          --密钥更新请求(7.3.5)
kup         [8]   CertRepMessage,           --密钥更新响应(7.3.6)
krr         [9]   CertReqMessages,          --密钥恢复请求(7.3.7)
krp         [10]  KeyRecRepContent,         --密钥恢复响应(7.3.8)
rr          [11]  RevReqContent,            --作废请求(7.3.9)
rp          [12]  RevRepContent,            --作废响应(7.3.10)
ccr         [13]  CertReqMessages,          --交叉认证请求(7.3.11)
ccp         [14]  CertRepMessage,           --交叉认证响应 (7.3.12)
ckuann      [15]  CAKeyUpdAnnContent,       --CA 密钥更新公告(7.3.13)
cann        [16]  CertAnnContent,           --证书公告(7.3.14)
rann        [17]  RevAnnContent,            --作废公告(7.3.15)
crlann      [18]  CRLAnnContent,            --CRL 公告(7.3.16)
pkiconf     [19]  PKIConfirmContent,        --确认(7.3.17)
nested      [20]  NestedMessageContent,     --嵌套消息(7.1.3)
genm        [21]  GenMsgContent,            --通用消息(7.3.19)
genp        [22]  GenRepContent,            --通用响应(7.3.20)
error       [23]  ErrorMsgContent,          --错误消息(7.3.21)
certConf    [24]  CertConfirmContent,       --证书确认(7.3.18)
pollReq     [25]  PollReqContent,           --轮询请求(7.3.22)
pollRep     [26]  PollRepContent            --轮询响应(7.3.22)
}
```

每一种类型将在 7.3 中描述。

7.1.3 PKI 消息保护

某些 PKI 消息需要保护完整性。(注意,如果使用非对称算法保护消息并且相应公钥部分已经被认证,那么消息的来源也可以被证明。另一方面,如果公钥部分未被认证,那么消息来源不能自动被证明,但可以通过带外方式被认证。)

protection 的结构如下:

```
PKIProtection ::= BIT STRING
```

计算 PKIProtection 时输入的是下面数据结构的 DER 编码:

```
ProtectedPart ::= SEQUENCE {
header PKIHeader,
body   PKIBody
}
```

在有些情况下,采用 PKIX 以外的其他保护方法保护消息,而不采用 PKIProtection 比特串(即省略这个可选的字段)。本标准允许选择这种方法。这种外部保护的例子包括使用 PKCS #7 [PKCS7]或 Security Multiparts [RFC1847] 对 PKIMessage 封装(当 PKIHeader 信息由外部机制安全的传输时,仅对省略 CHOICE 标签的 PKIBody 进行封装)。然而需要注意的是,许多这样的外部机制要求终端实体已经拥有一个公钥证书和/或一个唯一的 DN 和/或其他与基础结构相关的信息。这样,这些方法对于初始注册、密钥恢复或其他具有“初始引导”特性的过程来说就不合适。对这些情况,可能必须使用 PKIProtection。将来,当修改后的外部机制可以适用于具有“初始引导”特性的场景时,PKIProtection 将会变得很少或根本就不再使用。

根据环境不同,PKIProtection 比特可以包含一个消息认证码(MAC)或数字签名。只有下面的几种情况:

a) 共享密钥信息

在这种情况下，发送方和接收方共享（通过带外方法或前一个 PKI 管理操作建立的）密钥信 PKIProtection 将包含一个 MAC 值，protectionAlg 如下（参见附录 B）：

```
id-PasswordBasedMac OBJECT IDENTIFIER ::= {1 2 840 113533 7 66 13}
PBMParameter ::= SEQUENCE {
salt                OCTET STRING,
owf                 AlgorithmIdentifier,
--散列函数算法标识
iterationCount   INTEGER,
-- OWF 的应用次数
mac                 AlgorithmIdentifier
-- MAC 算法标识
}
```

在上面的 protectionAlg 中，在共享密钥的后面将追加上 salt 值。然后应用 iterationCount 次 OWF 算法，其中，追加了 salt 值的密钥作为第一次迭代的输入，对后续的每一次迭代，其输入都是前一次迭代的输出。最后一次迭代的输出（为了方便引用称作“BASEKEY”，其长度为“H”）用于形成对称密钥。如果 MAC 算法要求一个 K-比特的密钥而且 K <= H，那么密钥就取 BASEKEY 的前 K 比特。如果 K > H，那么 BASEKEY 的全部 H 比特将作为密钥的前 H 比特，OWF（“1” || BASEKEY）所得结果将作为密钥的下一 H 比特，OWF（“2” || BASEKEY）所得结果将作为密钥的再下一 H 比特，依此类推，直到得到所有的 K 比特为止。（这里，“N”代表数字 N 的 ASCII 字节编码，“||”代表串联。）

注：建议 PBMParameter 字段在一个交易的所有消息中（例如 ir/ip/certConf/pkiConf）保持不变，这样可以降低计算 PasswordBasedMac 的负荷。

b) 非对称密钥对

使用非对称密钥算法进行消息保护，下面以 DH 算法为例说明：

当发送者和接收者拥有相容的 DH 参数的 DH 证书时，为了保护消息，终端实体必须基于自己的 DH 私钥值和 PKI 消息接收方的 DH 公钥值产生一个对称密钥。PKIProtection 将包含由这个对称密钥计算出的 MAC 值，protectionAlg 如下：

```
id-DHBasedMac OBJECT IDENTIFIER ::= {1 2 840 113533 7 66 30}
DHBMParameter ::= SEQUENCE {
owf             AlgorithmIdentifier,
-- 散列函数算法 ID
mac             AlgorithmIdentifier
-- MAC 算法 ID
}
```

在上面的 protectionAlg 中，对 D-H 计算的结果应用 OWF。OWF 的输出（为了方便引用称作“BASEKEY”，其长度为“H”）用于形成对称密钥。如果 MAC 算法要求一个 K-比特的密钥而且 K <= H，那么密钥就取 BASEKEY 的前 K 比特。如果 K > H，那么 BASEKEY 的全部 H 比特将作为密钥的前 H 比特，OWF（“1” || BASEKEY）所得结果将作为密钥的下一 H 比特，OWF（“2” || BASEKEY）所得结果将作为密钥的再下一 H 比特，依此类推，直到得到所有的 K 比特为止。（这里，“N”代表数字 N 的 ASCII 字节编码，“||”代表串联。）

c) 签名

在这种情况下，发送方拥有一个签名密钥对，对 PKI 消息进行签名即可。PKIProtection 将包

含签名值，protectionAlg 为一种用于数字签名的算法。

d) 多重保护

当终端实体发送一个保护的 PKI 消息到 RA，RA 可以追加自己的保护（可以是 MAC 或签名，取决于 RA 和 CA 之间共享的信息和证书）后将消息转发到 CA。这种保护通过将终端实体发送的消息整个嵌套到一个新的 PKI 消息中实现。使用的结构如下：

NestedMessageContent ::= PKIMessages

7.2 公共数据结构

在定义 PKIBody 中的特定类型之前，我们先定义一些多处使用的数据结构。

7.2.1 被申请的证书内容

各种 PKI 管理消息都要求消息的发起者指明证书里存放的某些字段的值。CertTemplate 结构体使得 EE 或者 RA 可以尽可能地指定它们所希望申请到的证书里的内容。CertTemplate 结构体与证书的内容完全一致，但所有的字段都是可选的。

注：即使消息的发起者指明了它所申请的证书的所有内容，CA 机构仍然可以自由改动实际发放的证书的字段值。如果申请者不能接受被修改后的证书，那么申请者必须回送一个 certConf 消息，其中或者不包括该证书（通过一个 CertHash），或者包含该证书（通过一个 CertHash）但同时状态置为“rejected”。CertHash 和 certConf 消息的定义以及使用参见 7.3.18。

CertTemplate 语法见附录 D 及 RFC2511。

7.2.2 加密值

在 PKI 消息中发送加密值（在本标准中，加密值限于私钥或者证书）时采用 EncryptedValue 数据结构。

EncryptedValue 的语法见 RFC2511。

使用该数据结构要求创建者和期望的接收者分别都能够进行加密和解密操作。典型情况下，这意味着发送者和接收者拥有，或者能够产生共享的秘密密钥。

如果 PKIMessage 的接收者已经拥有用于解密的私钥，则 encSymmKey 字段可以包含用接收者的公钥加密的会话密钥（session key）。

7.2.3 PKI 消息的状态编码和失败信息

所有的响应消息都包含某些状态信息。下面定义了相应的值：

```
PKIStatus ::= INTEGER {
accepted                (0),
-- 表示得到了所要求的数据
grantedWithMods         (1),
-- 得到的数据与所要求的类似，但申请者有责任确定有无差别
rejection               (2),
-- 无法得到的数据，在该消息的其他地方有更多的信息
waiting                 (3),
-- 请求的包体尚未被处理，期望稍后将获取结果（注意：对具有该状态的响应，恰当的处理方式
-- 可以是使用 3.3.22 中定义的轮询请求/响应 PKIMessages；使用底层的传输层轮询机制
-- 也是一种可行的方法）
revocationWarning       (4),
-- 本消息包含一个即将作废的警告信息
revocationNotification  (5),
-- 通知已经作废
```

国家图书馆专用

```
keyUpdateWarning    (6)
-- 在密钥更新请求消息中 oldCertId 指示的密钥已经更新
}
```

响应者可以使用下列语法以提供有关失败状况的更多信息。

```
PKIFailureInfo ::= BIT STRING {
-- 因为多种情况可能导致失败,所以在需要时可以加入更多的代码
badAlg                (0),
-- 不可识别或者不支持的算法标识符
badMessageCheck       (1),
-- 完整性检查失败(例如,签名验证不成功)
badRequest            (2),
-- 不允许或不支持的交易
badTime               (3),
-- 根据本地策略,请求中的 messageTime 与系统时间不够接近
badCertId             (4),
-- 无法找到与提供的条件匹配的证书
badDataFormat         (5),
-- 提交的数据格式错误
wrongAuthority        (6),
-- 请求中指明的权威机构与本响应的创建者不同
incorrectData         (7),
-- 申请者的数据错误(用于公证服务)
missingTimeStamp      (8),
-- 在要求存在时戳的时候没有提供(根据策略要求)
badPOP                (9),
-- 拥有证明失败
}
PKIStatusInfo ::= SEQUENCE {
status          PKIStatus,
statusString    PKIFreeText     OPTIONAL,
failInfo        PKIFailureInfo  OPTIONAL
}
```

7.2.4 证书标识

CertId 数据结构用于鉴别特定的证书。

CertId 的语法见 RFC2511。

7.2.5 “带外”根证书公钥

每个根 CA 必须能够通过一些“带外”方式来发布它的当前公钥。虽然这样的方法不在本文描述的范围之内,我们定义了支持这种方法的数据结构。

一般而言,有两种可能的方法:CA 机构直接发布它的自签名证书;或者可以通过目录服务器(或其他类似方式)获得自签名证书,同时 CA 发布自签名证书的哈希值用于(由使用者)在使用(CA 证书)之前进行完整性校验。

```
OOBCert ::= Certificate
```

该证书中的值域有如下限制:

国家图书馆专用

—— 证书必须是自签名的(即,签名必须可以用 SubjectPublicKeyInfo 中的值来验证);

—— 主体字段和签发者字段的值必须完全相同;

—— 如果主体字段为空,则主体可替换名和签发者可替换名扩展项必须同时存在,并且具有完全相同的值;

—— 所有其他扩展项的值必须符合自签名证书的要求(比如,主体和签发者的密钥标识符必须完全相同)。

```
OOBCertHash ::= SEQUENCE {
hashAlg      [0]   AlgorithmIdentifier     OPTIONAL,
certId       [1]   CertId                  OPTIONAL,
hashVal            BIT STRING
-- hashVal 是对由 certID 所标识的自签名证书进行运算得出的值
}
```

散列(哈希)值的目的在于:任何(通过带外方式)安全获取到散列(哈希)值的用户都能验证该 CA 的自签名证书。

7.2.6 归档选项

请求者可以用 PKIArchiveOptions 结构体来指明它们希望 PKI 归档某私钥。

PKIArchiveOptions 的语法见 RFC2511。

7.2.7 发布信息

请求者可以用 PKIPublicationInfo 结构体来指明它们希望 PKI 发布证书。

PKIPublicationInfo 的语法见 RFC2511。

7.2.8 拥有证明结构体

如果是为签名密钥对申请证书(即申请一个验证证书),则可以使用 POPOSigningKey 结构体来证明对签名私钥的拥有。

POPOSigningKey 的语法见附录 D 及 RFC2511,但注意本标准中 POPOSigningKeyInput 存在下面的语义约定:

```
POPOSigningKeyInput ::= SEQUENCE {
authInfo             CHOICE {
sender               [0] GeneralName,
-- 取自 PKIHeader (仅当发送者的身份鉴别已通过某种方式(如以前发布的、当前处于有效状态的
-- 证书中的 DN)建立之后使用)
publicKeyMAC             PKMACValue
-- 发送者当前没有已鉴别通过的通用名(GeneralName)存在时使用;publicKeyMAC 包含一个
-- 基于口令的针对公钥的 DER 编码的 MAC 值(使用 PKIHeader 中的 protectionAlg 算法标识)
},
publicKey            SubjectPublicKeyInfo
-- 取自 CertTemplate
}
```

另一方面,如果为加密密钥对申请证书(即申请一个加密证书),则可以使用下列三种方式之一来证明对解密私钥的拥有。

a) 通过在证书请求中包含加密后的私钥(在 PKIArchiveOptions 控制结构体中)。

b) CA 不直接返回证书,而是返回加密后的证书(即,返回的证书用一个随机产生的对称密钥加密,而对称密钥本身用证书请求中包含的公钥来加密)—— 这是 6.3.2 中提到的“间接”方式。

国家图书馆专用

终端实体通过使用源自该对称密钥的密钥计算 PKIConfirm 消息的 MAC 值，来向 CA 机构证明它拥有解密私钥。（当 PKIMessage 消息中包含不止一个 CertReqMsg 证书请求，并且每个 CertReqMsg 使用不同的对称密钥时，MAC 计算使用源自所有这些对称密钥的一个密钥。）计算 MAC 的操作使用 7.1 定义的 PasswordBasedMac AlgId。

c) 让 EE 在 CertReqMessages 和 CertRepMessage 之间参与挑战－响应协议（使 POPODecKeyChall 和 POPODecKeyResp 消息；参照下文）—— 这是 6.3.2 中提到的“直接”方式。（这种方式典型的应用环境是 RA 验证 POP，然后代表终端实体向 CA 发送认证请求。在这种情况下，CA 相信 RA 在为终端实体申请证书之前已经正确地验证了 POP。）完整的协议如下所示（注意 req′不一定要将 req 封装为嵌套消息）：

```
EE          RA          CA
---- req ---->
<--- chall ---
---- resp --->
                 ---- req' --->
                 <--- rep -----
                 ---- conf --->
                 <--- ack -----
<--- rep -----
---- conf --->
<--- ack -----
```

这一协议显然比前面的方式 2 中给定的 3 次交换要长，但它允许本地注册机构（LRA）介入，并使得 POP 验证通过之前并不真正生成证书。

如果为密钥协商密钥对（KAK）申请证书，则 POP 的验证可以使用对加密密钥对验证的三种方式中的任何一种，但要做下列变化：（1）前面描述的第二种方式中的括号中的文本变为：（即，证书由根据 CA 的 KAK 私钥和认证请求中的公钥生成的对称密钥加密）；（2）下面 Challenge 结构描述中的 challenge 字段的第一个括号中的文本改为：（使用 PreferredSymmAlg（见 7.3.18）和一个对称密钥，该密钥基于 CA 的 KAK 私钥和认证请求中的公钥而产生）。此外，如果 CA 已经拥有一个为终端实体所知的 D-H 证书，POP 可以使用 RFC2511 中定义的 POPOSigningKey 结构体（其中，alg 字段取值为 DH-BasedMAC，签名字段取值为 MAC）作为证明 POP 的第四种方式。

用于解密私钥 POP 的挑战－响应消息定义如下。注意，挑战－响应交换通过 PKIHeader 中的 transactionID 以及对 PKIMessage 的保护（MAC 或签名）与前面的认证请求消息（以及后续的认证响应和确认消息）相关联。

```
POPODecKeyChallContent ::= SEQUENCE OF Challenge
-- 每个加密密钥认证请求对应一个 Challenge（次序与请求出现在 CertReqMessages 中的次序保持
-- 一致）
Challenge ::= SEQUENCE {
owf                 AlgorithmIdentifier OPTIONAL,
-- 在第一个 Challenge 中必须存在；而在 POPODecKeyChallContent 随后的 Challenge 中可以省略
--（如果省略的话，则使用前一个 Challenge 中的 owf 字段。）
witness             OCTET STRING,
-- 对随机产生的整数 A 应用单向函数（owf）的结果。注意每个 Challenge 都必须使用不同的整数
```

国家图书馆专用

```
challenge                    OCTET STRING
-- 加密后的 Rand(使用认证请求中的公钥加密),Rand 定义如下
-- Rand ::= SEQUENCE {
-- int INTEGER,
-- 上面提到的随机产生的整数 A
-- sender GeneralName
-- 发送者的名称(与 PKIHeader 中的相同)
-- }
}
POPODecKeyRespContent ::= SEQUENCE OF INTEGER
-- 每个加密密钥认证请求对应一个整数 (这些整数的顺序与请求出现在 CertReqMessages 中的顺
-- 序保持一致)。收到的整数(上面提到的 A)被返回给相应 Challenge 的发送者
```

7.3 与操作相关的数据结构

7.3.1 初始化请求

一个初始化请求消息的 PKIBody 包含的是一个 CertReqMessages 数据结构体,这个结构体详细说明了所请求的证书。通常,可以为每一个申请的证书提供下列模板字段:SubjectPublicKeyInfo、KeyId 和 Validity (详细信息参见附录 B)。此消息用于实体在 PKI 体系中的最初初始化。

CertReqMessages 的语法见附录 D 及 RFC2511。

7.3.2 初始化响应

一个初始化响应消息的 PKIBody 包含的是一个 CertRepMessage 数据结构体,这个结构体对应每个证书请求均包括一个 PKIStatusInfo 字段、一个主题证书及可能存在的一个私钥(这个私钥通常由会话密钥加密,而会话密钥本身由 protocolEncrKey 来加密)。

CertRepMessage 的语法见 7.3.4。如果 PKI 消息的保护方式是"共享秘密信息"(见 7.1.3),则 caPubs 字段内的任何证书都可以被消息发起者作为根 CA 证书来直接信任。

7.3.3 注册/认证请求

一个注册/认证请求消息的 PKIBody 包含的是一个 CertReqMessages 数据结构体,这个结构体详细说明了所请求的证书。当已存在于 PKI 体系中实体想获取额外的证书时使用此消息。

CertReqMessages 的语法见附录 D 及 RFC2511。

作为可选择,PKIBody 也可以包含一个 CertificationRequest 结构体(此结构体在[PKCS10]的 ASN.1 结构 CertificationRequest 部分有完全详细的说明)。当需要与早期系统进行一些交互操作时,申请签名密钥证书的请求中可以使用该结构体,但不是绝对必要的情况下建议不使用此结构体。

7.3.4 注册/认证响应

一个认证响应消息的 PKIBody 包含的是一个 CertRepMessage 数据结构体,这个结构体包含了与每个证书请求对应的状态值,同时根据情况还可能包含一个 CA 的公钥、失败信息、一个主题证书和一个加密后的私钥。

```
CertRepMessage ::= SEQUENCE {
caPubs [1] SEQUENCE SIZE (1..MAX) OF Certificate OPTIONAL,
response SEQUENCE OF CertResponse
}
CertResponse ::= SEQUENCE {
certReqId                INTEGER,
-- 使用此项使响应与请求对应 (若相对应的请求中未指明 certReqId ,则此项应填-1)
status                   PKIStatusInfo,
```


国家标准馆专用

```
certifiedKeyPair        CertifiedKeyPair OPTIONAL,
rspInfo                 OCTET STRING OPTIONAL
-- 类似于 RFC2511 中 CertReqMsg 结构中为 regInfo 定义的 id-regInfo-utf8Pairs 字符串
}
CertifiedKeyPair ::= SEQUENCE {
certOrEncCert           CertOrEncCert,
privateKey      [0]     EncryptedValue OPTIONAL,
-- 参见 RFC2511 中编码的说明
publicationInfo [1]     PKIPublicationInfo OPTIONAL
}
CertOrEncCert ::= CHOICE {
    certificate         0]  Certificate,
    encryptedCert       [1] EncryptedValue
}
```

在每个 CertResponse 中 failInfo(在 PKIStatusInfo 中)和 certificate(在 CertifiedKeyPair 中)两个字段只能出现一个(取决于状态)。在某些状态值(例如:waiting)的情况下,两者都不会出现。

给定了 EncryptedCert 和相关的解密私钥就可以获得证书。这种方式的目的是允许 CA 返回证书值,但只有预期的接收者才能够获得实际的证书。这种方法的好处在于 CA 在无法证明请求者就是拥有相应私钥的终端实体的情况下依然可以返回证书(注意:直到 CA 收到返回的 PKIConfirm 消息后才能证明)。因此,在 POP 验证出错时 CA 并不是必须作废该证书。

7.3.5 密钥更新请求内容

密钥更新请求使用 CertReqMessages 语法结构。通常,可以为每一个要更新的密钥提供下列模板字段:SubjectPublicKeyInfo, KeyId 及 Validity。这个消息用于请求更新已存在的证书(未作废且未过期)。该操作有时被称作"证书更新"操作。所谓更新证书就是使用新的公钥或以原有公钥的证书替换现有的证书。

CertReqMessages 的语法见附录 D 及 RFC2511。

7.3.6 密钥更新响应内容

密钥更新响应使用 CertRepMessage 语法结构。该响应与初始化响应相同。

CertRepMessage 的语法见 7.3.4。

7.3.7 密钥恢复请求内容

密钥恢复请求的语法与初始化请求中的 CertReqMessages 相同。通常,可以为一个签名公钥申请证书时提供下列模板字段:SubjectPublicKeyInfo 和 KeyId(进一步的信息见附录 B)。

CertReqMessages 的语法见附录 D 及 RFC2511。注意:如果需要密钥历史,请求者必须在请求消息中加入协议加密密钥。

7.3.8 密钥恢复响应内容

密钥恢复响应使用下列语法。对于某些状态值(如:waiting),所有的可选项均不会出现。

```
KeyRecRepContent ::= SEQUENCE {
status          PKIStatusInfo,
newSigCert      [0]     Certificate             OPTIONAL,
caCerts         [1]     SEQUENCE SIZE (1..MAX) OF Certificate OPTIONAL,
keyPairHist     [2]     SEQUENCE SIZE (1..MAX) OF
CertifiedKeyPair OPTIONAL
}
```

7.3.9 撤销请求内容

当请求撤销一个证书(或多个)时,使用下列数据结构。请求者的名字出现在 PKIHeader 结构体中。

```
RevReqContent ::= SEQUENCE OF RevDetails
RevDetails ::= SEQUENCE {
certDetails          CertTemplate,
-- 允许请求者尽可能多的提交请求撤销的证书的资料
--(例如:在无法获得序列号的情况下)
crlEntryDetails      Extensions          OPTIONAL
-- 所请求的 crlEntryExtensions
}
```

7.3.10 撤销响应内容

撤销响应是对上述消息的响应,生成该响应后将发送给撤销请求者。(可以向请求作废证书的拥有者发送一个撤销公告消息)。

```
RevRepContent ::= SEQUENCE {
status               SEQUENCE SIZE (1..MAX) OF PKIStatusInfo,
-- 与 RevReqContent 中发送的顺序相同
revCerts     [0]     SEQUENCE SIZE (1..MAX) OF CertId OPTIONAL,
-- 撤销请求的 IDs (与 status 的顺序相同)
crls         [1]     SEQUENCE SIZE (1..MAX) OF CertificateList OPTIONAL
-- 结果 CRL (可能不止一个)
}
```

7.3.11 交叉认证请求内容

交叉认证请求与一般认证请求使用的语法相同(CertReqMessages),但具有下列限制:密钥对必须由请求 CA 产生且私钥一定不能发送给响应 CA。这个请求也可以用于子 CA 请求父 CA 为其签发的证书。

CertReqMessages 的语法见附录 D 及 RFC2511。

7.3.12 交叉认证响应内容

交叉认证响应与一般认证响应使用的语法相同(CertRepMessage),其约束是不会发送加密的私钥。

CertRepMessage 的语法见 7.3.4。

7.3.13 CA 密钥更新公告内容

当 CA 更新了自己的密钥对后,可以使用下列数据结构宣布这一事件。

```
CAKeyUpdAnnContent ::= SEQUENCE {
oldWithNew           Certificate, -- 用新私钥签名的旧证书
newWithOld           Certificate, -- 用旧私钥签名的新证书
newWithNew           Certificate,-- 用新私钥签名的新证书
}
```

7.3.14 证书公告

这个结构体可以用于公告证书的存在。

```
CertAnnContent ::= Certificate
```

注:此消息适用于没有其他证书发布方法的情况;对于利用 X.500 目录服务发布证书等情况不建议使用此消息。

国家图书馆专用

7.3.15 撤销公告

当 CA 已撤销或即将撤销一个证书时，可以公告这一事件(或将要来临的事件)。

```
RevAnnContent ::= SEQUENCE {
status              PKIStatus,
certId              CertId,
willBeRevokedAt     GeneralizedTime,
badSinceDate        GeneralizedTime,
crlDetails          Extensions OPTIONAL
-- 额外的 CRL 细节(如:crl 序列号、作废原因、位置等)
}
```

CA 可以使用这个消息来警告(或通知)一个证书持有者它的证书将要(或已经)被撤销。这主要用于撤销请求来自于非证书持有者的情况。

willBeRevokedAt 字段包含的是一个新的条目将加入到相应 CRLs 的时间。

7.3.16 CRL 公告

当 CA 签发了一个新的 CRL(或一批 CRL)时，可以使用以下数据结构宣布这一事件。

```
CRLAnnContent ::= SEQUENCE OF CertificateList
```

7.3.17 PKI 确认内容

这个消息在协议数据交换中作为最后的 PKIMessage。由于 PKIHeader 中已包含了所有必需的消息，所以它的内容为空。

```
PKIConfirmContent ::= NULL
```

7.3.18 PKI 通用消息内容

```
InfoTypeAndValue ::= SEQUENCE {
infoType            OBJECT IDENTIFIER,
infoValue           ANY DEFINED BY infoType OPTIONAL
}
-- InfoTypeAndValue 的内容包含但不限于下列值(更多的细节见后续章条，
-- 语法描述见附录 F)：
-- { CAProtEncCert      = {id-it 1}, Certificate                    }
-- { SignKeyPairTypes   = {id-it 2}, SEQUENCE OF AlgorithmIdentifier }
-- { EncKeyPairTypes    = {id-it 3}, SEQUENCE OF AlgorithmIdentifier }
-- { PreferredSymmAlg   = {id-it 4}, AlgorithmIdentifier            }
-- { CAKeyUpdateInfo    = {id-it 5}, CAKeyUpdAnnContent             }
-- { CurrentCRL         = {id-it 6}, CertificateList                }
-- 其中，{id-it} = {id-pkix 4} = {1 3 6 1 5 5 7 4}，这个构造也可用于定义
-- 新的 PKIX 证书管理协议的请求和相应消息或通用目的消息(例如公告)
-- 以满足未来或特定环境的要求
GenMsgContent ::= SEQUENCE OF InfoTypeAndValue
-- EE,RA，或者 CA 都可以发送该消息(取决于消息的内容)。对于上面的给出的某些例子，GenMsg
-- 中 InfoTypeAndValue 的可选参数 infoValue 通常会被省略(即它只应用于相关的 GenRep 消息
-- 中)。接收方有权忽略它不能识别的对象标识符。如果消息由 EE 发往 CA，若结构内容为空的
-- 表明 CA 可以发送其希望发送的任意或所有信息
```

国家图书馆专用

7.3.19 PKI 通用响应消息

```
GenRepContent ::= SEQUENCE OF InfoTypeAndValue
-- 接收方可以忽略其不能识别的 OID
```

7.3.20 错误消息内容

```
ErrorMsgContent ::= SEQUENCE {
pKIStatusInfo          PKIStatusInfo,
errorCode              INTEGER          OPTIONAL,
-- 与实现相关的错误码
errorDetails           PKIFreeText      OPTIONAL
-- 与实现相关的错误描述
}
```

8 必需的 PKI 管理功能

8.1 根 CA 初始化

新建立的根 CA 要求能够产生自己的自签名证书，这个证书结构与根 CA 密钥更新后发布的“newWithNew”证书结构相同。

为了使 CA 自己的证书对的自签名证书对那些不通过带外方法来获得这个证书的终端实体有用，CA 要为它的公钥产生一个指纹。终端实体通过带外方法安全地获得这个指纹，然后就可以验证 CA 的证书以及证书里面的其他属性。

传递指纹的数据结构是 OOBCertHash。

8.2 根 CA 密钥更新

CA 密钥（与其他的所有密钥一样）的生命期是有限的，因此需要进行周期性的更新。CA 发布 newWithNew，newWithOld，和 oldWithNew 证书（见 6.4.1）用以帮助现存的终端实体将拥有的自签名 CA 证书（oldWithOld）安全地转变为新的自签名 CA 证书（newWithNew），同时也帮助新的即将获得 newWithNew 证书的终端实体安全地取得 oldWithOld 证书来验证现存的数据。

8.3 下级 CA 初始化

从 PKI 管理协议的角度来看，下级 CA 的初始化与终端实体的初始化是一样的。唯一的区别在于下级 CA 必须同时产生初始的撤销列表。

8.4 CRL 产生

新建立的 CA（它能发布 CRL）在发布证书之前必须首先产生空的 CRL。

8.5 PKI 信息请求

当 PKI 实体（CA，RA，或 EE）希望获得关于 CA 当前状态的信息时，它可以向 CA 发送对这种信息的请求。CA 必须向请求者提供至少请求者要求的所有请求的信息。如果某些信息不能提供，则必须给请求者返回错误。

如果使用 PKI 消息来请求和提供 PKI 相应的信息，那么请求必须使用 GenMsg 消息，响应使用 GenRep 消息，错误使用 Error 错误消息。这些消息使用基于共享秘密信息的 MAC（如：PasswordBasedMAC）或者其他认证方法（如果终端实体拥有证书）来进行保护。

在证书撤消请求中使用共享秘密信息的安全机制参见附录 E。

8.6 交叉认证

请求者 CA 作为交叉证书的主体，而响应者 CA 作为交叉证书的发布者。

请求者 CA 在发起交叉认证操作之前必须已建立并在运行中。

交叉认证方案基本上是单向的操作；也就是说，当成功时，这个操作会导致一个新的交叉证书的产生。如果需要产生“双向”的交叉证书，那么每一个 CA 都要发起一个交叉认证操作（或者使用另一种方

国家图书馆专用

案)。

这个方案适合于下面两种情况:两个 CA 已经验证过对方的签名(它们有共同的信任点);或者能对认证请求起源进行带外验证。详细描述:

交叉认证由作为响应者的 CA 发起。作为响应者的 CA 管理员鉴别它想交叉认证的 CA,同时由响应者 CA 设备产生授权码。响应者的 CA 管理员用带外方法把授权码传递给请求者的 CA 管理员。请求者的 CA 管理员在请求者 CA 处输入授权码来发起在线交换。

授权码用于认证和完整性目的验证。它是这样实现的:利用授权码产生对称密钥,然后使用此对称密钥对所有的交换消息产生消息认证码(MAC)。(如果 CA 可以通过某些方法检索和验证请求的公钥,那么认证也可以通过使用签名来代替 MAC。)

请求者 CA 用一个新的随机数(请求者随机数)产生交叉认证请求(ccr)以发起交换。然后,请求者 CA 给响应者 CA 发送 ccr 消息。消息字段使用基于认证码的 MAC 来防止修改。

收到 ccr 消息之后,响应者 CA 验证消息和 MAC,保存请求者随机数,并且产生自己的随机数(响应者随机数)。然后产生(如果需要就存档)一个新的包含请求者 CA 公钥的请求者证书,同时用响应者 CA 签名私钥进行签名。响应者 CA 用交叉认证响应(ccp)消息进行响应。消息中的字段使用基于认证码的 MAC 来防止修改。

收到 ccp 消息之后,请求者 CA 验证消息(包括收到的随机数)和 MAC。请求者 CA 用 certConf 消息进行响应。消息中的字段使用基于认证码的 MAC 来防止修改。请求者 CA 把请求者的该(交叉认证)证书写入证书库为今后的证书路径构建提供帮助。

收到 certConf 消息之后,响应者 CA 验证消息和 MAC,同时使用 PKIConfirm 消息发回一个确认。它也会公布请求者的(交叉认证)证书为今后路径构建提供帮助。

ccr 消息必须包含一个"完整"的认证请求,也就是说,请求者 CA 必须规定好除了序列号的所有的字段(包括,例如:BasicConstraints 扩展)。ccp 消息应该包含响应者 CA 的验证证书—如果存在,请求者 CA 必须验证这个证书(例如,通过带外机制)。

可以设想一种更为简单的非交互式的交叉认证模型,在这个模型中发布者 CA 从证书库中获得主体 CA 的公钥,并通过带外机制进行验证,然后产生和公布交叉证书而不需要主体 CA 的参与。在很多环境中这个模型是非常合理的,但是因为它不需要任何协议消息交换,所以详细描述不在本标准的范围之内。

交叉认证的消息结构见附录 C。

8.7 终端实体初始化

和 CA 一样,终端实体也需要初始化。终端实体的初始化至少有两个步骤:

—— 获得 PKI 信息;

—— 对根 CA 公钥的带外验证。

(可能还会有别的步骤,包括检索信任条件信息和对别的 CA 公钥的带外验证。)

8.7.1 获得 PKI 信息

需要获得的信息:

—— 当前根 CA 的公钥;

——(假如认证 CA 不是根 CA)包括适当的撤销列表的从根 CA 到认证 CA 以及适当的撤销列表的认证路径;

—— 认证 CA 支持的每一种相关应用的算法和算法变量。

为了能产生一个成功的认证请求可能需要一些附加的信息(例如:支持的扩展或者 CA 策略信息)。然而,为了简单化我们不强制终端实体通过 PKI 消息获得这些信息。最终的结果有可能导致一些认证请求的失败(例如,如果终端实体想产生自己的加密密钥但是 CA 不允许)。

需要的信息可以按照 8.5 所描述的来获得。

8.7.2 根 CA 密钥的带外验证

终端实体必须安全地持有根 CA 的公钥。为了保证安全，可以通过一些安全的“带外”方法给终端实体提供 CA 的证书指纹。这样终端实体就可以安全地使用 CA 的证书。详见 8.1。

8.8 证书请求

经过初始化的终端实体可以在任何时候(为了任何目的)请求一个额外的证书。这个请求使用认证请求(cr)消息。如果终端实体已经拥有一对签名密钥(带有相应的验证证书)，那么这个 cr 消息就使用实体的数字签名来保护。CA 用 CertRepMessage 返回新的证书(如果请求成功)。

8.9 密钥更新

当密钥对过期时相关的终端实体就会请求密钥更新——也就是说，它可以请求 CA 为新密钥对发布新的证书(或者，在特定情况下，给相同的密钥对一个新证书)。这个请求使用密钥更新请求(kur)消息(参考，在某些情况下和“证书更新”操作一样)。如果终端实体已经拥有了签名密钥对(带有相应的验证证书)，那么这个消息就使用实体的数字签名来保护。CA 用密钥更新响应(kup)消息返回新的证书(如果请求成功)，这个消息在语法构成上与 CertRepMessage 一样。

9 传输

下面指定的传输协议允许 EE、RAs 以及 CAs 在它们之间传送 PKI 消息。如果 PKI 消息进行适当的保护(即如果每一个消息都按规定使用了可选的 PKIProtection)，那么在这一层上，不要求使用特殊的安全机制。

9.1 基于文件的协议

包含 PKI 消息的文件必须仅仅包含一个 PKI 消息的 DER 编码，即文件中不能有额外的头信息和尾信息。这样的文件可以用于传输 PKI 消息，如使用 FTP 协议。

9.2 直接基于 TCP 的管理协议

下面的简单的基于 TCP 的协议可以用于传输 PKI 消息。这一协议适用于 EE 或 RA 启动一个 transaction 并可以通过轮询来获得结果的情况。

如果事务处理由 PKI 实体(RA 或 CA)启动，那么 EE 要么必须提供一个侦听进程，要么由启动方提供一个轮询参考数以允许它从 PKI 管理部件处获得 PKI 消息。

本协议假设在 RA 或 CA 上有一个监听进程在知名端口 829 上侦听 PKI 消息。通常情况下，启动者绑定到该端口并为给定的 transactionID 提交初始的 PKI 消息。响应者向请求方返回一个 PKI 消息和/或一个参考数(用于其后的轮询以获得真正的 PKI 响应消息)。

如果对于给定的请求将产生多个 PKI 响应消息(假设请求的某些部分比其他部分处理的快)，那么将返回一个新的轮询参考数。

当最后一个 PKI 响应消息已被启动者获得后，响应方将不提供新的轮询参考数。

事务处理的启动者向接收者发送一个“直接基于 TCP 的 PKI 消息”。接收者返回一个类似的消息。

一个“直接基于 TCP 的 PKI 消息”由下列各部分组成：

length (32-bits), flag (8-bits), value (defined below)

length —— 包含该消息的其他部分的字节数(即“value”的字节数加 1)。本协议中所有的 32-bit 值都采用网络序。

消息名	标志	值
pkiMsg	'00'H	DER-encoded PKI message
-- PKI 消息		
pollRep	'01'H	polling reference (32 bits), time-to-check-back (32 bits)
-- 无准备好的 PKI 消息时返回的 poll 响应；使用其中的 polling		

-- reference 值(在估计的时间间隔后)再次查询

pollReq '02'H polling reference (32 bits)

-- 请求对最初的 PKI 消息的响应

negPollRep '03'H '00'H

-- 再没有 polling 响应(即,transaction 完毕)

partialMsgRep '04'H next polling reference (32 bits),
time-to-check-back (32 bits),
DER-encoded PKI message

-- 对最初的 PKI 消息的部分响应,加上用于得到下一部分响应的新
-- polling reference(和估计的时间间隔值)

finalMsgRep '05'H DER-encoded PKI message

-- 对最初的 PKI 消息的最终(可能是唯一的)响应

errorMsgRep '06'H human readable error message

-- 检测到错误时产生这一消息(如,接收到一个不存在的或已经结束
-- 的 polling reference)

当要传输 PKIConfirm 消息时(总是从请求方发送到响应方),发送一个 pkiMsg 消息并返回一个 negPollRep 消息。

可能发生的消息序列为:

a) EE 发送 pkiMsg 消息,接收到的响应为 pollRep、negPollRep、partialMsgRep 或 finalMsgRep 中的一种;

b) EE 发送 pollReq 消息,接收到的响应为 negPollRep、partialMsgRep、finalMsgRep 或 errorMsgRep 中的一种。

参数"time-to-check-back"是一个 32-bit 的整数,定义为从 1970 年 1 月 1 日午夜 12 点开始经过的秒数。这是 EE 应当发送下一个 pollReq 的估计时间。

9.3 基于 E-mail 的管理协议

本条指定了一种通过 E-mail 传送第 7 章中描述的用于协议交换的 ASN.1 编码的消息的方法。

一个简单的 MIME 对象定义如下:

Content-Type: application/pkixcmp

Content-Transfer-Encoding: base64

<< ASN.1 DER 编码的 PKIX-CMP 消息, base64 编码>>

这个 MIME 对象可以使用普通的 MIME 处理机制(引擎)进行发送和接收,提供了一种利用 E-mail 传输 PKIX-CMP 消息的方法。具体实现可能希望识别并使用"application/x-pkixcmp"MIME 类型(本文档的早期版本中指定)以支持向后的兼容性。

MIME 对象的描述参见附录 G。

9.4 基于 HTTP 的管理协议

本条指定了一种通过 HTTP 协议传送上面定义的消息的 DER 编码的方法。

一个简单的 MIME 对象定义如下:

Content-Type: application/pkixcmp

<< ASN.1 DER 编码的 PKIX-CMP 消息 >>

这个 MIME 对象可以通过 WWW 链接使用普通的 HTTP 处理机制(引擎)进行发送和接收,提供了一种利用浏览器-服务器方式传输 PKIX-CMP 消息的方法。具体实现可能希望识别并使用"application/x-pkixcmp"MIME 类型以支持向后的兼容性。

MIME 对象的描述参见附录 G。

国家图书馆专用

附 录 A
（资料性附录）
RA 存在的原因

RA 存在的原因有技术和组织两方面。技术原因包括下述几点：

—— 如果使用硬件令牌，那么有可能不是所有的终端实体都有对其令牌进行初始化的设备，RA 设备就必须包括相应的功能（这是一个策略问题）；

—— 有些终端实体不能发布证书，同样 RA 可以帮助实现这个功能；

—— 当终端实体不能发布签名的撤销请求（如果完全丢失了密钥对）时，RA 就代表与其关联的终端实体进行发布。

RA 存在的组织原因如下：

—— 把功能集中在 RA 设备中比给所有的终端实体提供功能要有效得多（特别是需要用到特殊的令牌初始化设备时）；

—— 在组织内建立 RA 可以减少 CA 的数量，这在有些情况下是很重要的；

—— RA 可以更好地鉴别人们的“电子”名字，特别是当 CA 在物理上远离终端实体的情况下；

—— 对于很多应用，在适当的位置已经存在了管理结构，因此很容易产生 RA 角色（这可能不适用于 CA）。

国家图书馆专用

附 录 B
（规范性附录）
必选的 PKI 管理消息结构

本附录包含了遵循本标准的实现必须支持的 PKI 消息的详细描述（参见第 8 章）。

在以下 PKI 管理操作中使用的 PKIMessage 的结构如下：

—— 初始的注册/认证；

—— 基本认证方案；

—— 证书请求；

—— 密钥更新。

B.1 大纲结构解释的通用规则

a) 当个人单独的大纲描述中没有可选（OPTIONAL）或缺省（DEFAULT）字段时，相关的消息中也不能有（例如：接收者可以以语法错误为由来丢弃包含这些字段的消息）。如果必须强制字段具备一些很明显的值，那么文中也不会特别提及（例如：在本标准中 pvno 总是为 2）。

b) 当结构中不止一个消息时，将分别进行描述。

c) PKIMessage 结构的 algorithmIdentifiers 将分别描述。

d) 存在一种“特殊”的 X.500 DN 被称为“NULL-DN”；这表明 DN 包含长度为 0 的 SEQUENCE OF RelativeDistinguishedNames（它的 DER 编码为‘3000’H）。

e) 当某字段需要 GeneralName，但是又没有合适的值时（例如：终端实体在知道它的名字之前产生了一个请求），GeneralName 就为 X.500 NULL-DN（例如：CHOICE 的 Name 字段含有 NULL-DN）。这种特殊的值也称为“NULL-GeneralName”。

f) 当描述某大纲没有给 GeneralName 规定值时，相关 PKIMessage 字段的值就为 NULL-GeneralName。这种情况通常发生在某些消息的 PKI Header 的 sender 字段。

g) 在字段命名发生歧义的时候，描述名字就使用“点”号（例如：“certTemplate.subject”表示 subject 字段在 certTemplate 字段中）。

h) 当“SEQUENCE OF types”作为消息的一部分时，我们使用以 0 为基数的数组符号来描述 SEQUENCE OF 中的字段。（例如：crm[0].certReq.certTemplate.subject 是指请求消息中第一个 certReqMsg 的子字段。）

i) 在第 B.4～B.6 章中所有的 PKI 消息交换需要由发送实体发送 certConf 消息，同时由响应实体发送 PKIConfirm 消息。在某些描述大纲中，由于包体为 NULL，而包头的内容参照上下文非常清楚，所以没有包括 PKIConfirm，因为它的体为 NULL，头内容在上下文中很清楚。对于 protectionAlg 可以使用任何认证方法（例如：如果知道共享的秘密信息可以使用基于口令的 MAC，或者签名）。

B.2 算法使用参数

表 B.1 包含了在 PKI 管理协议中使用的算法定义。

国家标准专用

表 B.1　PKI 管理协议使用的算法

Name:消息结构使用的标识符	Use:算法使用的原因和地方
MSG_SIG_ALG	用签名保护 PKI 消息
MSG_MAC_ALG	用 MAC 保护 PKI 消息
SYM_PENC_ALG	使用带外方式发布对称密钥时对称加密终端实体的私钥
PROT_ENC_ALG	用于加密在 PKIMessages 中传输的(用于加密的对称密钥中的)私钥的非对称算法
PROT_SYM_ALG	用于加密私钥的比特位(这种类型的密钥使用 PROT_ENC_ALG 进行加密)的对称加密算法

B.3　所有权证明大纲结构

当需要证明拥有请求的证书中与验证公钥相应的签名私钥时,使用 POP 字段(在 ProofOfPossession 结构中的 pop 字段中的 signature 字段)。

字段	值	说明
algorithmIdentifier	MSG_SIG_ALG	这个证明只允许采用签名保护
signature	present	使用 MSG_SIG_ALG 计算比特位

证明拥有请求的证书中与加密公钥相应的解密私钥时不按照这种结构;而是使用 certConf 消息的 CertHash 字段来代替。

在带内认证请求协议中,不是每一个 CA/RA 都需要所有权证明(签名密钥,解密密钥或密钥一致密钥)。(如何进行 POP 基本上是一个策略问题,每一个 CA 都会在公布的 Policy OID 和 Certification Practice Statement 中表示清楚)。然而,本标准要求 CA/RA 实体把 POP(通过某些方法)作为认证过程的一部分。所有的终端实体都必须能够提供 POP(例如:PKIX-CMP 协议的组件必须支持)。

B.4　初始的注册/认证(基本认证方案)

(未初始化的)终端实体向 CA 请求(第一个)证书。当 CA 返回包含有证书的消息时,终端实体会发送证书确认。CA 再发回 PKIConfirm,关闭交易。所有的消息都进行了认证。

本方案允许终端实体请求对本地产生公钥的认证(典型地为签名密钥)。终端实体也可以选择请求集中产生并且认证另一个密钥对(典型地为加密密钥对)。

请求认证只能用于一个本地产生的公钥(对于更多的公钥,需要使用独立的 PKIMessage)。

终端实体必须支持对与本地产生公钥相关联的私钥的所有权证明。

前提:

—— 终端实体可以验证基于带外方法的 CA 签名;

—— 终端实体和 CA 共享对称 MACing 密钥。

消息流:

步骤	终端实体				PKI
1	format ir				
2		->	ir	->	
3					handle ir
4					format ip
5		<-	ip	<-	
6	handle ip				
7	format certConf				

8 -> certConf ->

9 handle certConf

10 format PKIConf

11 <- PKIConf <-

12 handle PKIConf

在本大纲中，我们要求终端实体在一个 PKIMessage 中包含所有的（例如：一个或者两个）CertReqMsg，同时 PKI（CA）必须产生一个包含完整响应 PKIMessage（例如：在请求发生进行了相应请求，并且支持集中密钥产生密钥的情况下，终端实体在一个 PKIMessage 中还要包括 OPTIONAL 可选的次第二对密钥对）。为了简化，我们要求这些消息必须是最后一个（例如：不使用“waiting”状态值）。

终端实体与 CA/RA 有带外交互作用。这个交易建立了共享秘密、referenceNumber 参考号码和用于证书模板中 sender 和 subject name 的可选的（OPTIONALLY）distinguished name 可识别名称 DN。建议共享秘密至少长 12 个字符。

ir：

字段	值
recipient	CA name
-- 被要求产生证书的 CA 的名字	
protectionAlg	MSG_MAC_ALG
-- 这个请求只允许 MAC 保护，基于	
-- 初始的认证密钥	
senderKID	referenceNum
-- 索引号，由 CA 预先发给	
-- 终端实体（同时还有 MACing 密钥）	
transactionID	present
-- 特定执行的值，对终端实体有意义	
--［如果该值在 CA 中已经使用，那么要发送由 CA 产生的拒绝消息］	
senderNonce	present
-- 128（伪）随机比特	
freeText	any valid value
body	ir（CertReqMessages）
-- 仅支持一个或两个 CertReqMsg	
-- 如果请求更多的证书，那么请求就要求被打包在独立的 PKIMessages 中	
CertReqMsg	one or two present
--见下面的详细资料细节，注意：crm[0]表示第一个（这是必须有的）	
-- crm[1]表示第二个（这是可选的，并且用于请求集中产生的密钥）	
crm[0]. certReq. certReqId	fixed value of zero
-- 消息中模板的索引	
crm[0]. certReq. certTemplate	present
-- 必须包含对象公钥值，否则就不受强制。其他的（值）不受限制	
crm[0]. pop. POPOSigningKey	optionally present if public key from crm[0]. certReq. certTemplate is a signing key
-- 在交换中可能要求所有权证明 POP（见第 B. 3 章）	
crm[0]. certReq. controls. archiveOptions	optionally present

国家图书馆专用

-- 终端实体可以请求获得本地产生被归档的私钥
crm[0]. certReq. controls. publicationInfo optionally present
-- 终端实体可以要求公布结果证书
crm[1]. certReq. certReqId fixed value of one
-- 消息中模板的索引
crm[1]. certReq. certTemplate present
crm[1]. certReq. controls. protocolEncrKey present
-- [object identifier MUST be PROT_ENC_ALG]
-- 如果 CA 支持集中产生密钥，CA 将用这个短期的非对称加密密钥
--（由终端实体产生）来加密（用对称密钥来加密）CA 为终端实体产生的私钥
-- 代表终端实体来加密（用于加密的对称密钥）CA 产生的私钥
crm[1]. certReq. controls. archiveOptions optionally present
crm[1]. certReq. controls. publicationInfo optionally present
protection present
-- 使用 MSG_MAC_ALG 计算比特位

ip：

字段	值
sender	CA name
-- 产生消息的 CA 的名字	
messageTime	present
-- CA 产生消息的时间	
protectionAlg	MS_MAC_ALG
-- 这个本响应只允许 MAC 保护	
senderKID	referenceNum
-- 索引号，由 CA 预先发给	
-- 终端实体（同时还有 MACing 密钥）	
transactionID	present
-- 相应 ir 消息中的值	
senderNonce	present
-- 128（伪）随机比特	
recipNonce	present
-- 相应 ir 消息中 senderNonce 的值	
freeText	any valid value
body	ip (CertRepMessage) contains exactly one response for each request

-- PKI（CA）适当地对一个或者两个请求的响应
-- crc[0] 表示第一个（一直存在）；crc[1]表示
-- 第二个（仅存在于 ir 消息包含两个请求并且
-- CA 支持集中地密钥产生）
crc[0]. certReqId fixed value of zero
-- 必须包含对相应 ir 消息中第一个请求的响应

国家图书馆专用

crc[0]. status. status	present, positive values allowed: "accepted", "grantedWithMods" negative values allowed: "rejection"
crc[0]. status. failInfo	present if and only if crc[0]. status. status is "rejection"
crc[0]. certifiedKeyPair	present if and only if crc[0]. status. status is "accepted" or "grantedWithMods"
certificate	present unless end entity's public key is an encryption key and POP is done in this in-band exchange
encryptedCert	present if and only if end entity's public key is an encryption key and POP done in this in-band exchange
publicationInfo	optionally present
-- 表明证书已经发布(由 CA 决定它的存在)	
crc[1]. certReqId	fixed value of one
-- 必须包含对相应 ir 消息中第二个请求的响应	
crc[1]. status. status	present, positive values allowed: "accepted", "grantedWithMods" negative values allowed: "rejection"
crc[1]. status. failInfo	present if and only if crc[0]. status. status is "rejection"
crc[1]. certifiedKeyPair	present if and only if crc[0]. status. status is "accepted" or "grantedWithMods"
certificate	present
privateKey	present (see Appendix D)
publicationInfo	optionally present
-- 表明证书已经发布(由 CA 决定它的存在)	
protection	present
-- 使用 MSG_MAC_ALG 计算比特位	
extraCerts	optionally present
-- CA 可以给终端实体提供额外附加的其他的证书	

certConf:

字段	值
sender	present
-- 与 ir 中的相同	

国家图书馆专用

recipient　　CA name
-- 被要求产生证书的 CA 的名字
transactionID　　present
-- 相应 ir 和 ip 消息中的值
senderNonce　　present
-- 128（伪）随机 比特
recipNonce　　present
-- 相应 ip 消息中 senderNonce 的值
protectionAlg　　MSG_MAC_ALG
-- 这个消息只允许 MAC 保护。MAC 是
-- 基于 EE 和 CA 共享的初始的 auth'n 密钥
senderKID　　referenceNum
-- 索引号，由 CA 预先发给
-- 终端实体（同时还有 MACing 密钥）
body　　certConf
-- 见 7.1.2 中 certConf 字段内容
-- 注意：如果发送了加密和签名的证书，那么
-- 需要两个 CertStatus 结构
protection　　present
-- 使用 MSG_MAC_ALG 计算比特位

PKIConf：
字段　　值
sender　　present
-- 与 ip 中的相同
recipient　　present
-- certConf 中发送者的名字
transactionID　　present
-- 从 certConf 消息中得到值
senderNonce　　present
-- 128（伪）随机 比特
recipNonce　　present
-- certConf 消息中 senderNonce 的值
protectionAlg　　MSG_MAC_ALG
-- 这个消息只允许 MAC 保护
senderKID　　referenceNum
body　　PKIConf
protection　　present
-- 使用 MSG_MAC_ALG 计算比特位

B.5　证书请求

（已经初始化的）终端实体（由于某种原因）向 CA 请求证书。当 CA 返回包含有证书的消息时，终端实体会发送证书确认。CA 再发回 PKIConfirm，关闭交易。所有的消息都进行了认证。

国家图书馆专用

本大纲中的消息交互流程与第 B.4 章中的基本一样，但是有以下不同：

—— 发送者名字应该存在；

—— 在 request，response，certConfirm 和 PKIConfirm 消息中必须支持 MSG_SIG_ALG 的 protectionAlg(也要支持 MSG_MAC_ALG)；

—— senderKID 和 recipKID 仅在请求消息验证时才存在；

—— body 为 cr 或 cp；

—— body 可能包含一种或两种 CertReqMsg 结构；

—— 保护比特根据 protectionAlg 字段来计算。

B.6 密钥更新请求

(已经初始化的)终端实体向 CA 请求证书(用于更新密钥对和/或已经拥有的相应证书)。当 CA 返回包含有证书的消息时，终端实体会发送证书确认。CA 再发回 PKIConfirm，关闭交易。所有的消息都进行了认证。

本大纲中的消息交互流程与第 B.4 章中的基本一样，但是有以下不同：

—— 发送者名字应该存在；

—— 在 request，response，certConfirm 和 PKIConfirm 消息中必须支持 MSG_SIG_ALG 的 protectionAlg(也要支持 MSG_MAC_ALG)；

—— senderKID 和 recipKID 仅在请求消息验证时才存在；

—— body 为 kur 或 kup；

—— body 可能包含一种或两种 CertReqMsg 结构；

—— 保护比特根据 protectionAlg 字段来计算。

国家图书馆专用

附 录 C
（规范性附录）
可选的 PKI 管理消息结构

本附录包含了遵循本标准的实现可以支持的 PKI 消息的详细描述（必须要支持的消息 ——见第 8 章和附录 B）。

在以下 PKI 管理操作中使用的 PKIMessage 的结构大纲如下：

—— 根 CA 密钥更新；
—— 信息请求/响应；
—— 交叉认证请求/响应；
—— 使用外部实体证书的带内初始化；
—— 撤销请求；
—— 证书发布；
—— CRL 发布。

C.1 结构解释的通用规则

（与第 B.1 章相同。）

C.2 算法使用参数

（与第 B.2 章相同。）

C.3 自签名证书

自签名证书结构用于分发“根”CA 的公钥。采用以下三种方式中的一种（2.4 中有这些结构的用法描述）：

类型	功能
newWithNew	真正的自签名证书；所包含的公钥必须能用于验证签名（尽管这只提供了完整性，并且没有任何认证）
oldWithNew	用新的私钥对以前的根 CA 公钥签名
newWithOld	用以前的私钥对新的根 CA 公钥签名

C.4 根 CA 密钥更新

根 CA 更新密钥对。而后它将产生（通过某些传输机制）相关终端实体能够获得的 CA 密钥更新声明消息。并且不需要从终端实体那里获得确认消息。

ckuann message：

字段	值	注释
sender	CA name	CA 名称
body	ckuann(CAKeyUpdAnnContent)	
oldWithNew	必须存在	见第 C.3 章
newWithOld	必须存在	见第 C.3 章
newWithNew	必须存在	见第 C.3 章
extraCerts	可选	用于发布证书（例如使用新私钥签发的证书）

国家图书馆专用

C.5 PKI 信息请求/响应

终端实体发送 general 通用消息给 PKI，以请求随后的 PKI 管理操作中所需要的细节。RA/CA 用 general 通用通常的响应消息来响应。如果 RA 产生响应，那么它将简单地将从 CA 那里收到的消息向前传转发，可能还会在 PKIMessage 中的 extraCerts 字段中加上证书。本消息不需要从终端实体处获得确认消息。

消息流：

步骤	终端实体				PKI
1	format genm				
2		->	genm	->	
3					handle genm
4					produce genp
5		<-	genp	<-	
6	handle genp				

genM：

字段	值
recipient	CA name
-- 证书中 issuerAltName 扩展或 issuer 字段包含的 CA 的名字	
protectionAlg	MSG_MAC_ALG or MSG_SIG_ALG
-- 任何一种鉴别保护算法	
SenderKID	present if required
-- 在需要验证消息保护的情况下必须存在	
freeText	any valid value
body	genr (GenReqContent)
GenMsgContent	empty SEQUENCE
-- 所有请求的相关信息	
protection	present
-- 使用 MSG_MAC_ALG 或 MSG_SIG_ALG 计算比特位	

genP：

字段	值
sender	CA name
-- 产生消息的 CA 的名字	
protectionAlg	MSG_MAC_ALG or MSG_SIG_ALG
-- 任何一种鉴别保护算法	
senderKID	present if required
-- 在需要验证消息保护的情况下必须存在	
body	genp (GenRepContent)
CAProtEncCert	present (object identifier one of PROT_ENC_ALG), with relevant value
-- 在终端实体为 CA 加密信息时使用(例如，用于私钥恢复)	
SignKeyPairTypes	present, with relevant value

国家图书馆专用

-- CA 为对象公钥验证的签名算法标识符集合
EncKeyPairTypes　　present，with relevant value
-- CA 为对象公钥验证的加密/密钥一致算法标识符集合
PreferredSymmAlg　　present(object identifier one
　　of PROT_SYM_ALG)，with relevant value
-- CA 希望在随后的 PKI 消息(为了加密)中使用的对称算法
CAKeyUpdateInfo　　optionally present，with relevant value
-- CA 使用这个字段提供关于相关根 CA 密钥对的信息
--(注意这并不表示响应的 CA 为所讨论的根 CA)
CurrentCRL　　optionally present，with relevant value
-- CA 提供 CRL 的完整拷贝（例如，最可能完整）
protection　　present
-- 使用 MSG_MAC_ALG 或 MSG_SIG_ALG 计算比特位
extraCerts　　optionally present
-- 可用于给终端实体发送证书
-- RA 可以把它的证书加在这里

C.6　交叉认证请求/响应(单向)

单个交叉证书的产生(例如：不要立刻产生两个不同时产生两个)。请求 CA 可以通过使用 PKI-PublicationInfo 控制来选择由谁来负责发布响应 CA 产生的交叉证书。

前提：

—— 响应 CA 在处理请求之前能够验证请求的来源(可能请求要求带外方法)；

—— 请求 CA 在处理响应之前可以鉴别响应源的真实性(可能请求要求带外方法)。

PKIHeader 中的 generalInfo 字段决定是否使用证书确认和相应的服务器确认(见 3.1.1)。以下流程不要求支持任何一种确认。

消息流：

步骤	请求发起 CA				请求响应 CA
1	format ccr				
2		->	ccr	->	
3					handle ccr
4					produce ccp
5		<-	ccp	<-	
6	handle ccp				

ccr：

字段	值
sender	Requesting CA name 请求 CA
-- 产生消息的 CA 的名字	
recipient	Responding CA name 响应 CA
-- 被要求产生证书的 CA 的名字	
messageTime	time of production of message 产生消息的时间
-- 请求 CA 的当前时间	
protectionAlg	MSG_SIG_ALG

-- 这个请求只允许签名保护
senderKID　　present if required
-- 如果在要求对消息保护进行验证时需要则必须存在
recipKID　　present if required
-- 如果在要求对消息保护进行验证时需要则必须存在
transactionID　　present
-- 执行特定与实现相关的值，对请求 CA 有意义
--（如果已经在响应 CA 中使用，那么响应 CA 产生的一个拒绝消息）
senderNonce　　present
-- 128 比特（伪）随机 比特数
freeText　　任何有效值
body　　ccr（CertReqMessages）
　　只允许一个 CertReqMsg
-- 如果请求多重交叉证书，那么就必须打包在不同的 PKIMessage
certTemplate　　present
-- 后面是细节如下
version　　v1 or v3
--建议 v3
signingAlg　　present
-- 请求 CA 必须事先知道用什么算法来对证书签名
subject　　present
-- 仅在提议使用 subjectAltNames 扩展项时可以为 NULL-DN
validity　　present
-- 必须完全指定（例如两个字段都存在）
issuer　　present
-- 仅在提议使用 issuerAltNames 扩展项时可以为 NULL-DN
publicKey　　present
-- 需要鉴定认证的密钥（用于必须是一种签名算法的密钥）
extensions　　optionally present
-- 请求 CA 必须为要求出现在交叉证书中请求的所有扩展建议取值
POPOSigningKey　　present
-- 见第 B.3 章
protection　　present
-- 使用 MSG_SIG_ALG 计算比特位
extraCerts　　optionally present
-- 可以包含请求者想要包括的任何额外证书

ccp：
字段　　值
sender　　响应 CA 的名字
-- 产生消息的 CA 的名字
recipient　　发起请求 CA 的名字
-- 被要求产生证书的 CA 的名字

国家图书馆专用

messageTime	产生消息的时间
-- 响应 CA 的当前时间	
protectionAlg	MSG_SIG_ALG
-- 这个消息只允许签名保护	
senderKID	present if required
--如果在要求对消息保护进行验证的情况下时需要则必须存在	
recipKID	present if required
transactionID	present
-- 相应 ccr 消息中的值	
senderNonce	present
-- 128 比特(伪)随机数 比特	
recipNonce	present
-- 相应 ccr 消息中的 senderNonce	
freeText	任何有效值
body	ccp (CertRepMessage)
	只允许一个 CertResponse
-- 如果请求多个交叉证书,那么就要就必须打包在不同的 PKIMessage 中	
response	present
status	present
PKIStatusInfo. status	present
-- 如果 PKIStatusInfo. status 为下面两种情况之一	
-- accepted,或 grantedWithMods	
-- 那么 certifiedKeyPair 必须存在,同时 failInfo 缺省	
failInfo	present depending on
	PKIStatusInfo. status
-- 如果 PKIStatusInfo. status 为	
-- rejection	
-- 那么 certifiedKeyPair 缺省,同时 failInfo 必须存在	
-- 而且设置了相应的比特位	
certifiedKeyPair	present depending on
	PKIStatusInfo. status
certificate	present depending on
	certifiedKeyPair
-- 在发布之前由请求 CA 必须检查的实际证书的内容	
protection	present
-- 使用 MSG_SIG_ALG 计算的比特位	
extraCerts	optionally present
-- 可以包含响应者想要包括的任何额外证书	

C.7 使用外部身份证书进行带内初始化

(未初始化的)终端实体希望初始化为 CA/PKI 的一员。为了鉴别身份,它使用原来一个已经存在的由另一个(外部)CA,CA-X 发布签发的身份证书。在 CA-1 和 CA-X 之间必须已经建立信任关系,这

国家图书馆专用

样 CA-1 才能使验证 EE 的由 CA-X 签名的证书生效。另外，在 EE 的个人安全环境（PSE）中必须建立一些机制，以便允许终端实体可以鉴别和验证由 CA-1 签名的 PKIMessage（例如，PSE 包含为 CA-1 公钥发布签发的证书，这个证书由终端实体信任的另一个 CA 在带外验证技术基础上签名，这种信任建立在带外认证技术基础上）。

终端实体发送初始化请求来启动交易。当 CA-1 用包含新证书的消息响应时，终端实体返回证书确认。CA-1 发送 PKIConfirm 来结束交易。所有的消息都经过签名（EE 消息使用与外部身份证书中的公钥相对应的私钥来签名；CA-1 消息用与 EE 的 PSE 中信任的另一个证书公钥相对应的私钥来签名）。

消息交互流程与 B.4 中的一样，但是有以下不同：

—— EE 和 CA-1 不共享对称 MACing 密钥（在这些实体之间没有带外共享的秘密信息）；

—— ir 中发送者 sender 名字必须存在（并且与外部身份证书中对象 subject 名字一样）；

—— 在所有消息的 protectionAlg 中必须使用 MSG_SIG_ALG 的 protectionAlg；

—— 在 ir 的 extraCerts 字段中必须携带外部身份证书；

—— 不使用 senderKID 和 recipKID；

—— body 为 ir 或 ip；

—— 保护比特数根据 protectionAlg 字段来计算。

国家图书馆专用

附 录 D
（资料性附录）
请求消息行为说明

以下的定义来自 RFC2511。主要是为了使对请求消息中的行为更系统说明；另外，所有的语法和语义与 RFC2511 中的一样。

```
CertRequest ::= SEQUENCE {
    certReqId           INTEGER,
    certTemplate        CertTemplate,
    controls            Controls OPTIONAL }
AltCertTemplate ::= AttributeTypeAndValue

POPOSigningKey ::= SEQUENCE {
    poposkInput                 [0] POPOSigningKeyInput OPTIONAL,
    algorithmIdentifier         AlgorithmIdentifier,
    signature                   BIT STRING }

POPOPrivKey ::= CHOICE {
    thisMessage         [0] BIT STRING,
    subsequentMessage   [1] SubsequentMessage,
    dhMAC               [2] BIT STRING }
```

国家图书馆专用

附 录 E
（资料性附录）
使用“口令短语”

撤销请求必须合并采用合适的安全机制，包括恰当的验证，用于减少拒绝服务攻击发生成功的可能性。对请求的数字签名（本标准中在支持撤销请求的情况下必须支持）能够满足验证要求，但是在某些情况下需要有替代机制（例如，私钥已经不能访问，但是终端实体希望在重新认证另一个密钥对之前请求撤销）。为了满足这些情况，在本标准中，如果支持撤销请求并且在需要撤销之前请求者和响应者之间可以建立共享的秘密信息，那么要求必须支持对请求的 PasswordBasedMAC（以符合给定环境的本地安全策略）。

“Revocation Passphrase”是已经在某些环境中使用的一种机制，在这种机制中，一个足够平均的值（例如，相对长的 passphrase 而不是短的 password）在撤销之前就在且仅在实体和 CA/RA 之间共享，并且在以后用于验证撤销请求。

在本标准中，是否支持下列用于建立共享秘密信息（例如，revocation passphrase）的技术是可选的。在 CMP 消息中的准确使用如下：

—— OID 以及在 7.3.19 中指定的 OID 以及值可在任何时间在 GenMsg 消息中发送，或者也可以在任何时间任何一个 PKIMessage 的 PKIHeader 的 generalInfo 字段中发送。（特别地，EncryptedValue可以在 certConf 消息头中发送 EncryptedValue，certConf 该消息是用于确认已经接受了在初始化请求和证书请求消息中所请求的证书。）这表明表示 revocation passphrase由终端实体（例如即，encValue 字段的解密后的字节）选择给为相关的 CA/RA 选择的 revocation passphrase；另外，传输也使用适当的加密机制来完成秘密字符进行传输（因为 passphrase 使用 CA/RA 的 protocolEncryptionKey 加密）。

—— 如果 CA/RA 在 GenMsg 中接收到 revocation passphrase（OID 以及在 7.3.19 中指定的 OID 以及值），它必须构造并发送 GenRep 消息，这个消息包含了在 7.3.19 中指定 OID（值为空）。如果 CA/RA 在任何一个 PKIMessage 的 PKIHeader 的 generalInfo 字段中接收到 revocation passphrase，它必须包含在相应响应 PKIMessage 的 PKIHeader 的 generalInfo 字段中的包含该 OID（值为空）。如果 CA/RA 由于某种原因不能返回适当的响应消息，那么它必须返回状态为“rejection”的错误消息，同时可以给出选的一组 failInfo 原因 reason set。

—— EncryptedValue 的 valueHint 字段可以包含了一个密钥标识符（由实体选择，同时还带有 passphrase 本身）用于帮助以后获得对正确的 passphrase 的检索（例如，当撤销请求由实体构造并且由 CA/RA 接收）。

—— 撤销请求消息用 PasswordBasedMAC 来保护，用 revocation passphrase 作为密钥。如果合适的话，PKIHeader 中的 senderKID 字段可以包含 valueHint 中的值。

使用以上技术，revocation passphrase 可以在任何时候在不请求额外消息或者带外交互的情况下进行初始化建立和更新，而不要求额外消息或带外交互。例如，撤销请求消息本身（通过使用 revocation passphrase 作为密钥的 MAC 来保护和验证，使用 revocation passphrase 作为密钥）可以在 PKIHeader 中包含一个新的 revocation passphrase 以用于验证以后对实体的其他证书的撤销请求。在某些情况下，这种机制优于其他可能会在撤销请求消息中显示会暴露 passphrase 的机制，因为这可能会引起允许拒绝服务攻击，所以在这种情况下显示暴露的 passphrase 会被未授权的第三方用于验证对实体别的证书的撤销请求。然而，由于在请求消息中没有显示暴露 passphrase，因此不要求在产生撤销请求的时候总是更新 passphrase（也就是说，在不同的时候对不同证书的撤销请求的验证可以使用同一个

国家图书馆专用

passphrase)。

另外,以上技术可以在不使用签名的情况下依然对整个撤销请求消息提供强加密保护。简单显示通过暴露 revocation passphrase 来验证撤销请求的技术不对请求消息的字段提供加密保护(因此请求撤销一个证书的请求可能会被未授权的第三方修改,而请求撤销该实体的另一个证书)。

国家图书馆专用

附 录 F
（规范性附录）
“可编译”的 ASN.1 模块

```
PKIXCMP {iso(1) identified-organization(3) dod(6) internet(1)
    security(5) mechanisms(5) pkix(7) id-mod(0) id-mod-cmp2000(16)}

DEFINITIONS EXPLICIT TAGS ::=

BEGIN

EXPORTS ALL

IMPORTS

    Certificate, CertificateList, Extensions, AlgorithmIdentifier,
    UTF8String
            FROM PKIX1Explicit88 {iso(1) identified-organization(3)
            dod(6) internet(1) security(5) mechanisms(5) pkix(7)
            id-mod(0) id-pkix1-explicit-88(1)}

    GeneralName, KeyIdentifier
            FROM PKIX1Implicit88 {iso(1) identified-organization(3)
            dod(6) internet(1) security(5) mechanisms(5) pkix(7)
            id-mod(0) id-pkix1-implicit-88(2)}

    CertTemplate, PKIPublicationInfo, EncryptedValue, CertId,
    CertReqMessages
            FROM PKIXCRMF {iso(1) identified-organization(3)
            dod(6) internet(1) security(5) mechanisms(5) pkix(7)
            id-mod(0) id-mod-crmf(5)}
    -- 参见附录 D 中行为说明编码

    CertificationRequest
            FROM PKCS-10 {iso(1) member-body(2) us(840) rsadsi(113549)
            pkcs(1) pkcs-10(10) modules(1) pkcs-10(1)}
    -- (在 RFC 2986 中指定,使用 1993 ASN.1 语法和 IMPLICIT 标记)
    -- 作为选择，执行实现者可以在这个模型模块中直接包括 PKCS10 语法
    -- 这个模型的剩下其余部分包含本地定义的 OID 和结构

CMPCertificate ::= CHOICE {
```

国家图书馆专用

```
    x509v3PKCert          Certificate
}

-- 当比特在线与 X.509 标准对“证书”的定义兼容时，这个语法允许在本证书管理协议中使用将来
-- 的证书类型(比如 X.509 属性证书,WAP WTLS 证书或者别种证书),现在已经产生了对通用性
-- 支持的需求。对于没有预见到需要支持别的证书类型的执行实现,如果它们希望的话,可以在编
-- 译这个 ASN.1 模型模块之前注释掉以上的结构并且“不注释”以下的结构
--(注意:与没有进行这种该操作的执行实现的互操作性将不受这个改变的影响。)

-- CMPCertificate ::= Certificate

    PKIMessage ::= SEQUENCE {
        header          PKIHeader,
        body            PKIBody,
        protection      [0] PKIProtection OPTIONAL,
        extraCerts      [1] SEQUENCE SIZE (1..MAX) OF CMPCertificate OPTIONAL
}

PKIMessages ::= SEQUENCE SIZE (1..MAX) OF PKIMessage

PKIHeader ::= SEQUENCE {
    pvno                INTEGER     { cmp1999(1), cmp2000(2) },
    sender              GeneralName,
    -- 标识发送者
    recipient           GeneralName,
    -- 标识预期的接收者
messageTime       [0] GeneralizedTime         OPTIONAL,
    -- 产生这个消息的时间
    --(当发送者认为该时间的传输合适时使用;即,时间对接收来说有意义)
    protectionAlg   [1] AlgorithmIdentifier     OPTIONAL,
    -- 用于计算保护比特位的算法
    senderKID       [2] KeyIdentifier           OPTIONAL,
    recipKID        [3] KeyIdentifier           OPTIONAL,
    -- 识别用于保护的特定密钥
    transactionID   [4] OCTET STRING            OPTIONAL,
    -- 标识交易;即在相关的请求、响应和确认消息中,该字段值相同
    senderNonce     [5] OCTET STRING            OPTIONAL,
    recipNonce      [6] OCTET STRING            OPTIONAL,
    -- 用于防止重放攻击的随机数,senderNonce 由消息的创建者赋值
    -- recipNonce 是由本消息的预期接收者先前插入到相关消息中的随机数
    freeText        [7] PKIFreeText             OPTIONAL,
    -- 可以用于表示上下文相关的说明(本字段主要由人工使用)
    generalInfo     [8] SEQUENCE SIZE (1..MAX) OF
```

国家图书馆专用

```
                    InfoTypeAndValue                OPTIONAL
    -- 可以用于表示上下文相关的说明(本字段不是供人工使用的)
}

PKIFreeText ::= SEQUENCE SIZE (1..MAX) OF UTF8String
    -- 按 UTF-8[RFC2279]编码的文本(注意:每一个 UTF8String 都可以包含
    -- 一个 RFC 1766/RFC 3066 语言标签,用于表示所包含文本属于哪一种语
    -- 言 —— 详情请参见 RFC2482)

PKIBody ::= CHOICE { -- message-specific body elements
    ir          [0] CertReqMessages,          -- 初始化请求
    ip          [1] CertRepMessage,           -- 初始化响应
    cr          [2] CertReqMessages,          -- 认证请求
    cp          [3] CertRepMessage,           -- 认证响应
    p10cr       [4] CertificationRequest,     -- 从[PKCS10]导入
    popdecc     [5] POPODecKeyChallContent,   -- pop 挑战
    popdecr     [6] POPODecKeyRespContent,    -- pop 响应
    kur         [7] CertReqMessages,          -- 密钥更新请求
    kup         [8] CertRepMessage,           -- 密钥更新响应
    krr         [9] CertReqMessages,          -- 密钥恢复请求
    krp         [10] KeyRecRepContent,        -- 密钥恢复响应
    rr          [11] RevReqContent,           -- 作废请求
    rp          [12] RevRepContent,           -- 作废响应
    ccr         [13] CertReqMessages,         -- 交叉认证请求
    ccp         [14] CertRepMessage,          -- 交叉认证响应
    ckuann      [15] CAKeyUpdAnnContent,      -- CA 密钥更新公告
    cann        [16] CertAnnContent,          -- 证书公告
    rann        [17] RevAnnContent,           -- 作废公告
    crlann      [18] CRLAnnContent,           -- CRL 公告
    pkiconf     [19] PKIConfirmContent,       -- 确认
    nested      [20] NestedMessageContent,    -- 嵌套消息
    genm        [21] GenMsgContent,           -- 通用消息
    genp        [22] GenRepContent,           -- 通用响应
    error       [23] ErrorMsgContent,         -- 错误消息
    certConf    [24] CertConfirmContent,      -- 证书确认
    pollReq     [25] PollReqContent,          -- 轮询请求
    pollRep     [26] PollRepContent           -- 轮询响应
}

PKIProtection ::= BIT STRING

ProtectedPart ::= SEQUENCE {
    header          PKIHeader,
```

国家图书馆专用

```
    body            PKIBody
}

id-PasswordBasedMac OBJECT IDENTIFIER ::= {1 2 840 113533 7 66 13}
PBMParameter ::= SEQUENCE {
    salt                    OCTET STRING,
    -- 为了减少拒绝服务攻击的可能性,执行实现可能会限制
    -- 这个串的可接受的串的长度以符合自己的环境中的取值
    -- 这样可以减少拒绝服务攻击
    owf                     AlgorithmIdentifier,
    -- 散列函数算法 ID
    iterationCount          INTEGER,
    -- OWF 的应用次数
    --为了减少拒绝服务攻击的可能性,实现执行可能会限制
    -- 这个整数的可接受的整数的大小以符合自己的环境中的取值
    -- 这样可以减少拒绝服务攻击
    mac                     AlgorithmIdentifier
    -- 单向算法 ID (例如, DES-MAC, Triple-DES-MAC [PKCS11]
}   -- 或者 HMAC [RFC2104, RFC2202])

id-DHBasedMac OBJECT IDENTIFIER ::= {1 2 840 113533 7 66 30}
DHBMParameter ::= SEQUENCE {
    owf                     AlgorithmIdentifier,
    -- 散列函数算法 ID
    mac                     AlgorithmIdentifier
    -- MAC 算法 ID (例如,DES-MAC, Triple-DES-MAC [PKCS11]
}   -- 或者 HMAC [RFC2104, RFC2202])


NestedMessageContent ::= PKIMessages

PKIStatus ::= INTEGER {
    accepted                (0),
    -- 表示得到了所要求的数据
    grantedWithMods         (1),
    -- 得到的数据与所要求的类似,但申请者有责任确定有无差别
    rejection               (2),
    -- 无法得到数据,在该消息的其他地方有更多的信息
    waiting                 (3),
    -- 请求的包体尚未被处理,期望稍后将获取结果
    revocationWarning       (4),
    -- 本消息包含一个即将作废的警告信息
    revocationNotification  (5),
```

国家图书馆专用

```
    -- 通知已经作废
    keyUpdateWarning        (6)
    -- 在密钥更新请求消息中 oldCertId 指示的密钥已经更新
}

PKIFailureInfo ::= BIT STRING {
    -- 因为多种情况可能导致失败,所以在需要时可以加入更多的代码
    badAlg                  (0),
    -- 不可识别或者不支持的算法标识符
    badMessageCheck         (1),
    -- 完整性检查失败(例如,签名验证不成功)
    badRequest              (2),
    -- 不允许或不支持的交易
    badTime                 (3),
    -- 根据本地策略,请求中的 messageTime 与系统时间不够接近
    badCertId               (4),
    -- 无法找到与提供的条件匹配的证书
    badDataFormat           (5),
    -- 提交的数据格式错误
    wrongAuthority          (6),
    -- 请求中指明的权威机构与本响应的创建者不同
    incorrectData           (7),
    -- 申请者的数据错误(用于公证服务)
    missingTimeStamp        (8),
    -- 在要求存在时戳的时候没有提供(根据策略要求)
    badPOP                  (9),
    -- 拥有证明失败
    certRevoked             (10),
        -- 证书已经被作废
    certConfirmed           (11),
        -- 证书已经被确认
    wrongIntegrity          (12),
        -- 无效的完整性,应当使用基于口令方式但采用了签名方式或反之
    badRecipientNonce       (13),
        -- 无效的接收者随机数,没有提供 RecipientNonce 或取值错误
    timeNotAvailable        (14),
        -- TSA 服务的时钟源无法得到
    unacceptedPolicy        (15),
        -- 请求的 TSA 策略不被 TSA 服务支持
    unacceptedExtension     (16),
        -- 要求的扩展项不被 TSA 服务支持
    addInfoNotAvailable     (17),
        -- 无法理解要求的附加信息或者该附加信息无法得到
```

国家图书馆专用

```
        badSenderNonce          (18),
            -- 无效的接收者信息,没有提供 SenderNonce 或取值错误
        badCertTemplate         (19),
            -- 无效的证书模板,或者缺少强制信息
        signerNotTrusted        (20),
            -- 消息的签发者不可知或者不被信任
        transactionIdInUse      (21),
            -- 交易标识符已经在用
        unsupportedVersion      (22),
            -- 消息版本号不支持
        notAuthorized           (23),
            -- 发送者未被授权发出前面的请求或者执行前面的操作
        systemUnavail           (24),
            -- 系统不可用,所以请求无法被处理
        systemFailure           (25),
            -- 系统失败,所以请求无法被处理
        duplicateCertReq        (26)
            -- 由于同样的证书已经存在,所以证书不能发放
    }

    PKIStatusInfo ::= SEQUENCE {
        status          PKIStatus,
        statusString    PKIFreeText     OPTIONAL,
        failInfo        PKIFailureInfo  OPTIONAL
    }

    OOBCert ::= CMPCertificate

    OOBCertHash ::= SEQUENCE {
        hashAlg     [0] AlgorithmIdentifier     OPTIONAL,
        certId      [1] CertId                  OPTIONAL,
        hashVal         BIT STRING
            -- hashVal 在相应证书的 subjectPublicKey 字段的 DER 编码之上计算
    }

    POPODecKeyChallContent ::= SEQUENCE OF Challenge
            -- 每个加密密钥认证请求对应一个 Challenge(

    Challenge ::= SEQUENCE {
        owf                 AlgorithmIdentifier OPTIONAL,
            -- 在第一个 Challenge 中必须存在;而在 POPODecKeyChallContent
            -- 随后的 Challenge 中可以省略(如果省略的话,则使用前一个
            -- Challenge 中的 owf 字段。)
```

国家图书馆专用

```
    witness          OCTET STRING,
        --对随机产生的整数 A 应用单向函数(owf)的结果
    challenge        OCTET STRING
        -- 加密后的 Rand(使用认证请求中的公钥加密),Rand 定义如下
        -- Rand ::= SEQUENCE {
        -- int INTEGER,
        -- 上面提到的随机产生的整数 A
        -- sender GeneralName
        -- 发送者的名称(与 PKIHeader 中的相同)
        -- }
}

POPODecKeyRespContent ::= SEQUENCE OF INTEGER
        -- 每个加密密钥认证请求对应一个整数
        -- 收到的整数(上面提到的 A)被返回给相应 Challenge 的发送者

CertRepMessage ::= SEQUENCE {
    caPubs          [1] SEQUENCE SIZE (1..MAX) OF CMPCertificate OPTIONAL,
    response            SEQUENCE OF CertResponse
}

CertResponse ::= SEQUENCE {
    certReqId               INTEGER,
    -- 使用此项使响应与请求对应(若相对应的请求中未指明 certReqId ,则此项应填-1)
    status                  PKIStatusInfo,
    certifiedKeyPair        CertifiedKeyPair        OPTIONAL,
    rspInfo                 OCTET STRING            OPTIONAL
    -- 类似于 RFC2511 中 CertReqMsg 结构中为 regInfo 定义的 id-regInfo-utf8Pairs 字符串
}

CertifiedKeyPair ::= SEQUENCE {
    certOrEncCert       CertOrEncCert,
    privateKey      [0] EncryptedValue      OPTIONAL,
    -- 参看 RFC2511 中编码的说明
    publicationInfo[1] PKIPublicationInfo OPTIONAL
}

CertOrEncCert ::= CHOICE {
    certificate     [0] CMPCertificate,
    encryptedCert   [1] EncryptedValue
}

KeyRecRepContent ::= SEQUENCE {
```

国家图书馆专用

```
        status                  PKIStatusInfo,
        newSigCert      [0] CMPCertificate                  OPTIONAL,
        caCerts         [1] SEQUENCE SIZE (1..MAX) OF
                                        CMPCertificate      OPTIONAL,
        keyPairHist     [2] SEQUENCE SIZE (1..MAX) OF
                                        CertifiedKeyPair    OPTIONAL
    }

    RevReqContent ::= SEQUENCE OF RevDetails

    RevDetails ::= SEQUENCE {
        certDetails         CertTemplate,
        -- 允许请求者尽可能多的说明请求作废的证书的相关资料
        --(例如:在无法获得序列号的情况下)
        crlEntryDetails     Extensions      OPTIONAL
        -- 请求的 crlEntryExtensions
    }

    RevRepContent ::= SEQUENCE {
        status              SEQUENCE SIZE (1..MAX) OF PKIStatusInfo,
        -- 与 RevReqContent 中发送的顺序相同
        revCerts            [0] SEQUENCE SIZE (1..MAX) OF CertId OPTIONAL,
        -- 作废请求的 IDs (与 status 的顺序相同)
        crls                [1] SEQUENCE SIZE (1..MAX) OF CertificateList OPTIONAL
        -- 结果 CRL (可能不止一个)
    }


    CAKeyUpdAnnContent ::= SEQUENCE {
        oldWithNew              CMPCertificate,
        -- 用新私钥签名的旧证书
        newWithOld              CMPCertificate,
        -- 用旧私钥签名的新证书
        newWithNew              CMPCertificate
        -- 用新私钥签名的新证书
    }

    CertAnnContent ::= CMPCertificate

    RevAnnContent ::= SEQUENCE {
        status                  PKIStatus,
        certId                  CertId,
        willBeRevokedAt         GeneralizedTime,
```

国家图书馆专用

```
    badSinceDate                GeneralizedTime,
    crlDetails                  Extensions OPTIONAL
    -- 额外的 CRL 细节(如:crl 序列号、作废原因、位置等)
}

CRLAnnContent ::= SEQUENCE OF CertificateList

CertConfirmContent ::= SEQUENCE OF CertStatus

CertStatus ::= SEQUENCE {
    certHash        OCTET STRING,
    -- 证书的 HASH,其算法与创建及验证证书签名的算法相同
    certReqId       INTEGER,
    -- 使确认与相应的请求/响应匹配
    statusInfo      PKIStatusInfo OPTIONAL
}

PKIConfirmContent ::= NULL

InfoTypeAndValue ::= SEQUENCE {
    infoType                OBJECT IDENTIFIER,
    infoValue               ANY DEFINED BY infoType      OPTIONAL
}
-- 实例 InfoTypeAndValue 的内容包括以下方面,但不是限制在这些方面,InfoTypeAndValue
-- 的内容包含但不限于下列值(对给定的环境,将本 ASN.1 模块的注释去掉并恰当地使用)
-- (不对 ASN.1 模型进行注释,同时对给定的环境进行恰当的使用)
-- id-it-caProtEncCert          OBJECT IDENTIFIER ::= {id-it 1}
-- CAProtEncCertValue             ::= CMPCertificate
-- id-it-signKeyPairTypes       OBJECT IDENTIFIER ::= {id-it 2}
-- SignKeyPairTypesValue          ::= SEQUENCE OF AlgorithmIdentifier
-- id-it-encKeyPairTypes        OBJECT IDENTIFIER ::= {id-it 3}
-- EncKeyPairTypesValue           ::= SEQUENCE OF AlgorithmIdentifier
-- id-it-preferredSymmAlg       OBJECT IDENTIFIER ::= {id-it 4}
-- PreferredSymmAlgValue          ::= AlgorithmIdentifier
-- id-it-caKeyUpdateInfo        OBJECT IDENTIFIER ::= {id-it 5}
-- CAKeyUpdateInfoValue           ::= CAKeyUpdAnnContent
-- id-it-currentCRL             OBJECT IDENTIFIER ::= {id-it 6}
-- CurrentCRLValue                ::= CertificateList
-- id-it-unsupportedOIDs        OBJECT IDENTIFIER ::= {id-it 7}
-- UnsupportedOIDsValue           ::= SEQUENCE OF OBJECT IDENTIFIER
-- id-it-keyPairParamReq        OBJECT IDENTIFIER ::= {id-it 10}
-- KeyPairParamReqValue           ::= OBJECT IDENTIFIER
-- id-it-keyPairParamRep        OBJECT IDENTIFIER ::= {id-it 11}
```

国家图书馆专用

```
-- KeyPairParamRepValue        ::= AlgorithmIdentifer
-- id-it-revPassphrase         OBJECT IDENTIFIER ::= {id-it 12}
-- RevPassphraseValue          ::= EncryptedValue
-- id-it-implicitConfirm       OBJECT IDENTIFIER ::= {id-it 13}
-- ImplicitConfirmValue        ::= NULL
-- id-it-confirmWaitTime       OBJECT IDENTIFIER ::= {id-it 14}
-- ConfirmWaitTimeValue        ::= GeneralizedTime
-- id-it-origPKIMessage        OBJECT IDENTIFIER ::= {id-it 15}
-- OrigPkiMessageValue         ::= PKIMessages
-- id-it-suppLangTags          OBJECT IDENTIFIER ::= {id-it 16}
-- SuppLangTagsValue           ::= SEQUENCE OF UTF8String
-- where
-- id-pkix OBJECT IDENTIFIER ::= {iso(1) identified-organization(3)
-- dod(6) internet(1) security(5) mechanisms(5) pkix(7)}
-- and
-- id-it OBJECT IDENTIFIER     ::= {id-pkix 4}
-- 这个构造也可用于定义新的 PKIX 证书管理协议的请求和相应消息或通用目的消息(例如公告)
-- 以满足未来或特定环境的要求

GenMsgContent ::= SEQUENCE OF InfoTypeAndValue

-- EE,RA,或者 CA 都可以发送该消息(取决于消息的内容)。对于上面给出的某些例子,GenMsg
-- 中 InfoTypeAndValue 的可选参数 infoValue 通常会被省略(即它只应用于相关的 GenRep 消息
-- 中)。接收方有权忽略它不能识别的对象标识符。如果消息由 EE 发往 CA,且结构内容为空,
-- 则表明 CA 可以发送其希望发送的任意或所有信息

GenRepContent ::= SEQUENCE OF InfoTypeAndValue
-- 接收方可以忽略其不能识别的 OID

ErrorMsgContent ::= SEQUENCE {
    pKIStatusInfo           PKIStatusInfo,
    errorCode               INTEGER          OPTIONAL,
-- 与实现相关的错误码
    errorDetails            PKIFreeText       OPTIONAL
-- 与实现相关的错误描述
}

PollReqContent ::= SEQUENCE OF SEQUENCE {
    certReqId               INTEGER
}

PollRepContent ::= SEQUENCE OF SEQUENCE {
    certReqId               INTEGER,
```

国家图书馆专用

```
        checkAfter                  INTEGER,
    -- 以秒为单位的时间
        reason                      PKIFreeText OPTIONAL
    }

END
```

国家图书馆专用

附 录 G
（资料性附录）
用于 E-MAIL 或者 HTTP 的 MIME 类型

MIME 类型

MIME 媒体类型名字：application

MIME 图标子类型名字：pkixcmp

编码注意事项：

内容会包含任意的八位字节值（对 PKI 消息的 ASN.1 DER 编码，与在 IETF PKIX Working Group specifications 中定义的一样）。采用 MIME e-mail 时需要 base64 编码；采用 HTTP 时不需要编码。

安全注意事项：

本 MIME 类型用于在 PKI 实体中传输 Public-Key Infrastructure(PKI) 消息。消息由 IETF PKIX 工作组来定义，用于建立和维护互联网 X.509 PKI。如果 PKI 消息本身得到了 PKIX 规范中所定义的保护，那么在这一级别中不要求采用特定的安全机制。

使用本媒体类型的应用软件：

应用使用证书管理、操作或者辅助的协议(IETF PKIX 工作组定义的)，通过 E-Mail 或者 HTTP 来发送 PKI 消息的应用软件。

国家图书馆专用

参考文献

[1] RFC2559 Internet X.509 Public Key Infrastructure Operational Protocols - LDAPv2(互联网 X.509 公共密钥基础设施操作协议- LDAPv2)

[2] RFC2585 Internet X.509 Public Key Infrastructure Operational Protocols: FTP and HTTP (互联网 X.509 公共密钥基础设施操作协议:FTP and HTTP)

[3] RFC2279 UTF-8, a transformation format of ISO 10646 (UTF-8,ISO 10646 一种转换格式)

[4] RFC1766/RFC3066 Tags for the Identification of Languages (语言识别的标签)

[5] RFC2482 Language Tagging in Unicode Plain Text (对 UNICODE 纯文本的语言标签)

[6] RFC1847 Security Multiparts for MIME: Multipart/Signed and Multipart/Encrypted(多用途的网际邮件扩充协议的部分安全:签名部分和加密部分)

[7] RFC2986 PKCS #10: Certification Request Syntax Specification Version 1.7 (证书请求文法规范 版本 1.7)

[8] RFC2104 HMAC: Keyed-Hashing for Message Authentication (HMAC:消息认证的加密散列)

[9] RFC2202 Test Cases for HMAC-MD5 and HMAC-SHA-1 (对 HMAC-MD5 和 HMAC-SHA-1 的测试案例)

[10] PKCS #10 Certification Request Syntax Standard (证书请求文法标准)

[11] PKCS #7 Cryptographic Message Syntax Standard (密码消息文法标准)

[12] PKCS#1 RSA Cryptography Standard (非对称密码加密的加密标准)

[13] PKCS#11 Cryptographic Token Interface Standard (密码令牌界面标准)

国家图书馆专用

中 华 人 民 共 和 国
国 家 标 准
**信息技术 安全技术 公钥基础设施
证书管理协议**
GB/T 19714—2005
*
中 国 标 准 出 版 社 出 版 发 行
北京西城区复兴门外三里河北街 16 号
邮政编码:100045
http://www.spc.net.cn
电话:63787337、63787447
2005 年 8 月第一版 2005 年 8 月电子版制作
*
书号:155066·1-23067

版权专有 侵权必究
举报电话:(010)68533533

GB/T 19714-2005