

中华人民共和国国家标准

GB/T 19714—2025

代替 GB/T 19714—2005

网络安全技术 公钥基础设施 证书管理协议

Cybersecurity technology—Public key infrastructure—
Certificate management protocol

2025-08-01 发布

2026-02-01 实施

国家市场监督管理总局 发布
国家标准化管理委员会

目 次

前言 III

1 范围 1

2 规范性引用文件 1

3 术语和定义 1

4 缩略语 2

5 通则 2

6 流程与消息结构 3

 6.1 终端与 RA 系统间协议 3

 6.2 RA 系统与 CA 系统间协议 4

 6.3 CA 系统与 KM 系统间协议 6

 6.4 CA 系统与资料库间协议 12

 6.5 终端与资料库间协议 14

附录 A (规范性) 必选的证书管理消息结构 22

 A.1 概述 22

 A.2 消息结构解释的通用规则 22

 A.3 算法使用参数 22

 A.4 所有权证明消息结构 23

 A.5 初始的注册/认证(基本认证方案) 23

 A.6 证书请求 28

 A.7 密钥更新请求 28

附录 B (资料性) 可选的证书管理消息结构 29

 B.1 概述 29

 B.2 结构解释的通用规则 29

 B.3 算法使用参数 29

 B.4 PKI 信息请求/响应 29

 B.5 使用外部身份证书进行初始化 30

附录 C (规范性) PKI 消息数据结构 32

 C.1 PKI 消息综述 32

 C.2 公共数据结构 36

 C.3 特定操作数据结构 41

附录 D (资料性) 版本协商 47

 D.1 通则 47

 D.2 与 GB/T 19714—2005 版本服务端对话的客户端 47

 D.3 接收 GB/T 19714—2005 版本消息的服务端 47

附录 E (资料性) 使用“口令短语” 48

附录 F (资料性) 证书管理协议 ASN.1 描述 49

参考文献 57

前 言

本文件按照 GB/T 1.1—2020《标准化工作导则 第 1 部分：标准化文件的结构和起草规则》的规定起草。

本文件代替 GB/T 19714—2005《信息技术 安全技术 公钥基础设施 证书管理协议》。与 GB/T 19714—2005 相比，除结构调整和编辑性改动外，主要技术变化如下：

- a) 更改了标准的范围(见第 1 章, 2005 年版的第 1 章)；
- b) 删除了 PKI 管理概述内容(见 2005 年版的第 5 章)；
- c) 删除了加密密钥私钥拥有证明(见 2005 年版的 6.3.2、7.2.8)和协商密钥私钥拥有证明相关内容(见 2005 年版的 6.3.3)；
- d) 删除了根 CA 的更新的相关内容(见 2005 年版的 6.4、8.2)；
- e) 删除了终端实体初始化和初始注册/认证相关的前提与限制(见 2005 年版的第 6 章)；
- f) 增加了“流程与消息结构”，描述各 PKI 组件间证书管理的流程与消息结构(见第 6 章)；
- g) 增加了协议支持的国家标准算法以及算法 OID(见附录 A 的表 A.1)；
- h) 增加了“transactionID”的相关描述(见附录 C 的 C.1.2.1)；
- i) 增加了协议版本字段取值的设置(见 C.1.2.1)；
- j) 增加了“隐式确认”数据结构(见 C.1.2.2)和“确认等待时间”数据结构(见 C.1.2.3)；
- k) 在 PKIHead 的 generalInfo 扩展项增加了对证书模板标识“certTemplateID”字段的支持(见 C.1.2.4)；
- l) 增加了“多重保护”相关描述(见 C.1.4)；
- m) 更改了加密值数据结构，修改为“SM2EnvelopedKey”(见 C.2.2, 2005 年版的 7.2.2)，协议中加密数据统一更改使用“SM2EnvelopedKey”(见 C.3.2、6.2.2, 2005 年版的 7.3.2、附录 E)；
- n) 增加了证书确认相关内容(见 C.1.3、C.3.15)；
- o) 增加了“轮询请求和响应”数据结构(见 C.3.19)；
- p) 增加了证书冻结和证书解冻请求与响应相关内容(见 C.1.3、C.3.20、C.3.21)；
- q) 增加了有关失败状况的更多信息(见 C.2.3)；
- r) 增加了利用 CertReqMessages 进行多个证书的申请与利用 CertRepMessage 进行多个证书的响应时，有多种实现方式的说明，明确了实现上可有多种选择(见 C.3.1 和 C.3.2)，明确了一种常见的实现方式(见 A.5)；
- s) 删除了交叉认证相关内容(见 2005 年版的 7.3.11、7.3.12、8.6)；
- t) 删除了 CA 初始化、终端实体初始化相关的 PKI 管理功能内容(见 2005 年版的第 8 章)；
- u) 删除了 CMP 协议的传输相关内容(见 2005 年版的第 9 章和附录 G)；
- v) 删除了“请求消息行为说明”(见 2005 年版的附录 D)，将其主要内容纳入附录 C 中(见 C.2.8、C.3.1)；
- w) 更改了证书管理协议 OID(见附录 F, 2005 年版的附录 F)。

请注意本文件的某些内容可能涉及专利。本文件的发布机构不承担识别专利的责任。

本文件由全国网络安全标准化技术委员会(SAC/TC 260)提出并归口。

本文件起草单位：北京数字认证股份有限公司、中国电子技术标准化研究院、西安西电捷通无线网络通信股份有限公司、博雅中科(北京)信息技术有限公司、长春吉大正元信息技术股份有限公司、上海市数字证书认证中心有限公司、华为技术有限公司、武汉大学、公安部第一研究所、亚数信息科技(上海)

有限公司、长扬科技(北京)股份有限公司、深圳市电子商务安全证书管理有限公司、陕西省信息化工程研究院、江南信安(北京)科技有限公司、郑州信大捷安信息技术股份有限公司、清华大学、格尔软件股份有限公司、广东省电子商务认证有限公司、同智伟业软件股份有限公司、北京时代新威信息技术有限公司、北京中关村实验室、浙江大华技术股份有限公司、工业和信息化部网络安全产业发展中心(工业和信息化部信息中心)、工信通(北京)信息技术有限公司、中国电子信息产业集团有限公司第六研究所、国网区块链科技(北京)有限公司、中科信息安全共性技术国家工程研究中心有限公司、数安时代科技股份有限公司、奇安信网神信息技术(北京)股份有限公司、中电科网络安全科技股份有限公司。

本文件主要起草人:高文华、高文举、李彦峰、李琴、王秉新、丁肇伟、王玉林、曾光、何德彪、胡光俊、林雪焰、夏鲁宁、夏冰冰、李向锋、傅大鹏、刘中、王月辉、张国强、李志勇、田玉存、李亮、郭燕飞、丁蓓蓓、邓晨、刘斌、赵婧、张紫薇、魏一才、赵华、沈志淳、张鑫、苏金岩、王志辉、郑会涛、赵晓荣、徐剑南、王彤、汪海洋、刘为华、贾珂婷、郑强、陈树乐、焦正坤、俞政臣、朱威儒、赵博鑫、张剑青、陈子雄、王进、王斌、王龙、杨珂、高振鹏、杜志强、安锦程、寇建波。

本文件及其所代替文件的历次版本发布情况为:

——2005年首次发布为GB/T 19714—2005;

——本次为第一次修订。

网络安全技术 公钥基础设施 证书管理协议

1 范围

本文件给出了公钥基础设施(PKI)中证书管理协议的结构和内容,规定了证书产生和管理所需要的协议消息格式。

本文件适用于公钥基础设施相关产品的研制,以及用于指导公钥基础设施相关产品的设计、开发和管理。

2 规范性引用文件

下列文件中的内容通过文中的规范性引用而构成本文件必不可少的条款。其中,注日期的引用文件,仅该日期对应的版本适用于本文件;不注日期的引用文件,其最新版本(包括所有的修改单)适用于本文件。

- GB/T 19713 网络安全技术 公钥基础设施 在线证书状态协议
- GB/T 20518—2018 信息安全技术 公钥基础设施 数字证书格式
- GB/T 25056 信息安全技术 证书认证系统密码及其相关安全技术规范
- GB/T 25069—2022 信息安全技术 术语
- GB/T 33560 信息安全技术 密码应用标识规范
- GB/T 35276—2017 信息安全技术 SM2 密码算法使用规范

3 术语和定义

GB/T 25069—2022 界定的以及下列术语和定义适用于本文件。

3.1

数字签名 digital signature

被数据单元的接收者用以确认数据单元的来源和完整性,达到保护数据,防止被伪造的目的,附加在数据单元上的一些数据,或是对数据单元所做的密码变换。

[来源:GB/T 25069—2022,3.576,有修改]

3.2

杂凑算法 hash-algorithm

基于杂凑函数实现的密码算法。

3.3

个人安全环境 personal security environment;PSE

终端用于安全存储证书及私钥的环境。

3.4

拥有证明 proof of possession;POP

终端用以证明自己拥有(即能使用)与其申请证书的公钥相对应的私钥。

3.5

PKI 组件 PKI component

用于进行证书相关活动的程序、模块或系统。

注：通常包括终端和用于提供服务的 PKI 管理组件。

3.6

PKI 管理组件 PKI management component

用于提供证书相关服务的程序、模块或系统。

3.7

PKI 消息 PKI message

证书管理协议中的一种特定数据结构。

4 缩略语

下列缩略语适用于本文件。

CA:证书认证机构(Certificate Authority)

CMP:证书管理协议(Certificate Management Protocol)

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

KM:密钥管理(Key Management)

LDAP:轻量级目录访问协议(Lightweight Directory Access Protocol)

LRA:本地注册机构(Local Registration Authority)

MAC:消息认证码(Message Authentication Code)

OCSP:在线证书状态查询协议(Online Certificate Status Protocol)

PKCS:公钥密码标准(Public Key Cryptography Standards)

PKI:公钥基础设施(Public Key Infrastructure)

POP:拥有证明(Proof of Possession)

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

RA:证书注册机构(Registration Authority)

5 通则

证书管理是对数字证书进行全过程管理的一系列过程。证书管理协议是证书认证系统组件之间实现证书管理的通信交互协议,证书认证系统的描述见 GB/T 25056—2018。将 GB/T 25056—2018 中 5.1 定义的各组件归并为下列 PKI 组件。

- 终端:通常表现为系统或模块。证书持有者通过其控制的终端申请数字证书并进行安全保管和使用,证书验证者通过其控制的终端验证证书。
- RA 系统:所具有的功能通常包括证书申请的受理和查验等,可分为本地注册管理系统和远程注册管理系统。在有些情况下,运行 RA 系统的机构会下设 LRA 使其承担 RA 系统的一部分功能,LRA 系统是 RA 系统的组成部分,作为证书发放受理点系统直接面向用户。RA 系统采用自己的私钥进行数字签名和身份认证。一个 RA 系统可与多个 CA 系统协同工作。
- CA 系统:实现证书的签发、交付等功能,该系统是受用户信任的权威机构的系统。CA 系统包括离线模块和在线模块,只有离线模块可使用 CA 系统的私钥。
- KM 系统:用于接受 CA 系统的密钥服务请求,并将处理结果返回给 CA 系统。CA 系统与 KM 系统间涉及的密钥服务按照请求-响应的步骤执行。请求由 CA 系统提出,发送到 KM 系

统；响应由 KM 系统发起，发送到 CA 系统。

——资料库：负责证书和 CRL 的存储与发布，以及为证书验证者终端提供证书及 CRL 的下载与证书的在线状态查询服务。在资料库中，证书验证者终端利用证书中标识的 CRL 地址，下载 CRL 并检验证书的有效性；按照 OCSP 实时在线查询证书的状态。

这些 PKI 组件间的联机交互通信协议见图 1。证书管理协议包括终端与 RA 系统、RA 系统与 CA 系统、CA 系统与 KM 系统、CA 系统与资料库、终端与资料库间的协议。上述协议的消息统称为证书管理消息。

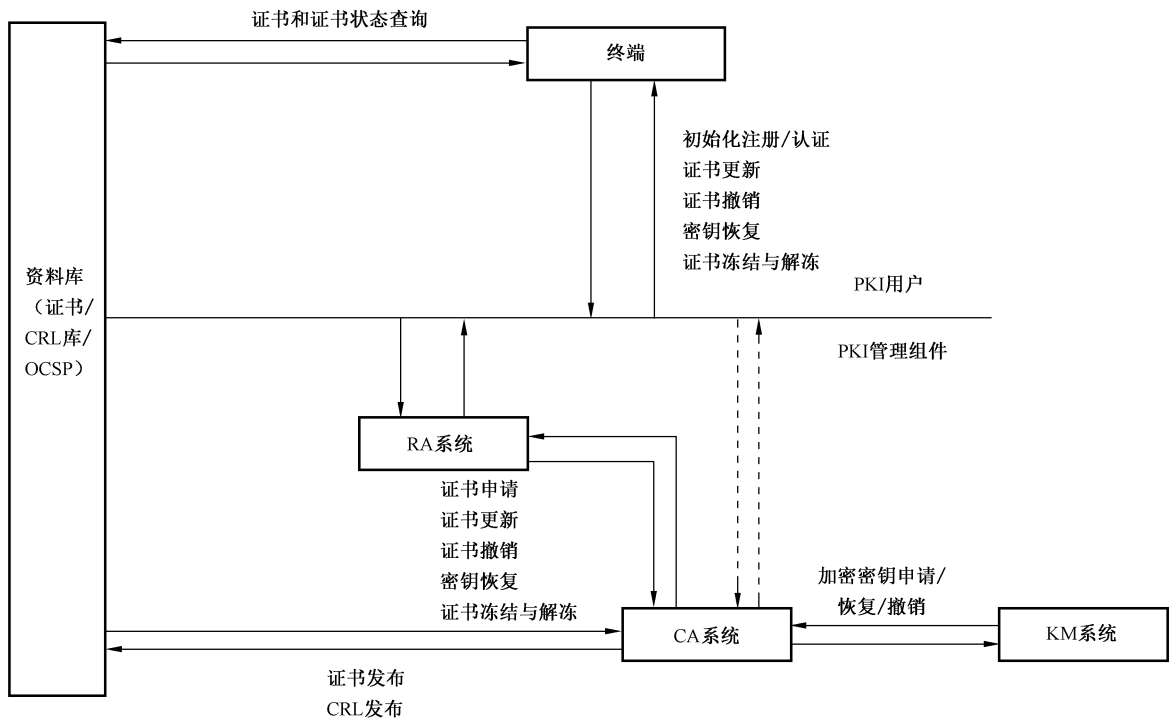


图 1 PKI 组件间的交互关系

应支持的证书管理消息的详细描述应符合附录 A，可支持的证书管理消息的详细描述见附录 B。

注：图 1 中虚线部分所涉及的协议流程及消息结构不在描述范围内，相关内容见 GB/T 19771 等相关标准。

6 流程与消息结构

6.1 终端与 RA 系统间协议

每一个终端在与 RA 系统进行交互之前，首先应通过一定的方式与 RA 系统建立安全连接、完成身份鉴别。终端与 RA 系统之间的身份认证可通过基于数字证书的身份鉴别来实现，也可通过共享密钥来完成。

终端与 RA 系统间的业务流程见图 2。终端与 RA 系统间的业务可能涉及终端初始注册/认证和证书申请、撤销、更新、冻结、解冻和密钥恢复等。

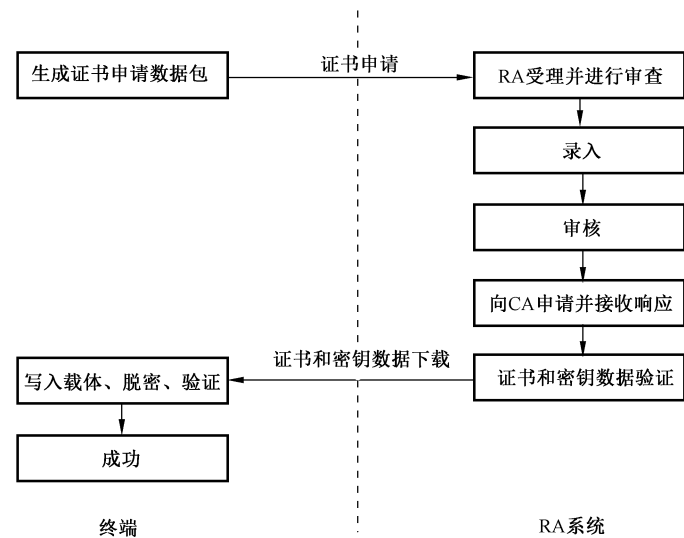


图 2 终端与 RA 系统间的业务流程

终端与 RA 系统间的交互可根据实际情况进行不同方式的实现。证书管理实现应符合附录 A 的规定。

6.2 RA 系统与 CA 系统间协议

6.2.1 协议概述

终端在 RA 系统注册并通过审核后,RA 系统与 CA 系统进行通信,实现证书的申请、撤销、更新、冻结、解冻和密钥恢复等功能,采用基于身份鉴别的安全通信协议,并进行双向身份鉴别,可防止中间人攻击,保证系统通信安全。RA 系统与 CA 系统间所有通信协议消息采用的数据结构应符合附录 C 的规定。

6.2.2 协议消息数据结构

6.2.2.1 证书申请协议

RA 系统与 CA 系统交互流程见图 3。



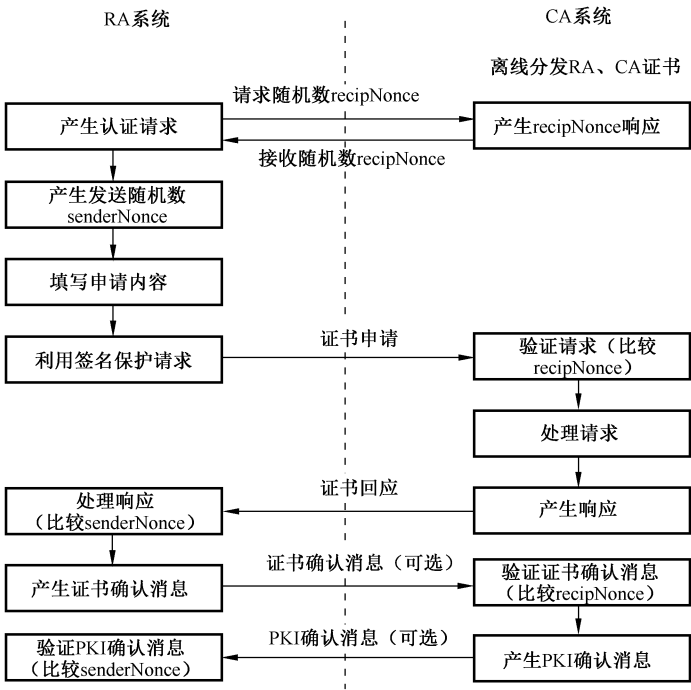


图3 RA系统与CA系统交互流程

如果CA系统收到请求后发现错误或其他原因不能正确回应,应直接返回错误消息,并中断此次连接。当验证确认消息失败时,如果新签发的证书已经发布或可通过其他方式获得,则CA系统应作废此证书。证书申请请求与响应数据结构如下。

- a) 证书申请请求数据结构。PKIBody 为 CertReqMessages,请求代码是 cr,其中指定了请求的 certificate(s)。通常情况下,对每一个请求都提供 SubjectPublicKeyInfo、KeyId 和 Validity 字段。具体数据结构应符合 C.3.3。CertReqMessages 中的部分字段说明如下。
- 注册信息 regInfo 仅包含与认证请求上下文相关的辅助信息。这些信息可包括用户的联系信息、计费信息或其他有助于完成认证请求的辅助信息。
 - 直接与证书内容相关的信息包含在 certReq 中。RA 系统计划放入证书内容中的信息可放入 regInfo。CRMF 中定义的注册信息包括: id-regInfo-utf8Pairs 和 id-regInfo-certReq。
 - CertRequest 中可能包含一个或多个与处理请求相关的控件。CRMF 中定义的控制信息包括: regToken、authenticator、pkiPublicationInfo、pkiArchiveOptions、oldCertID 及 protocolEncrKey。
 - CertTemplate 结构允许 RA 系统为其申请的证书指定任意多的字段。
- b) 证书申请响应数据结构。PKIBody 为 CertRepMessage,响应代码是 cp,对所请求的每一个证书,CertRepMessage 中都包含一个 PKIStatusInfo 字段,并可选地具有 CA 公钥、失败信息、申请者证书和加密的私钥(通常由一个会话密钥加密,而会话密钥本身由 protocolEncrKey 加密)。具体数据结构应符合 C.3.4。CertRepMessage 中的部分字段说明如下。
- 依赖于响应的状态,在每一个 CertResponse 中 PKIStatusInfo 的 failInfo 和 CertifiedKey-Pair 中的证书只能出现一个。对某些状态值(如 waiting),任何可选字段都不出现。
 - 若在 PKI 消息中发送加密的值(私钥或证书),使用 SM2EnvelopedKey 类型的数据结构。

6.2.2.2 证书撤销协议

RA 系统请求作废一个或几个证书时,使用下面的数据结构。RA 系统的名字在 PKIHeader 结构

中。流程见 6.2.2.1, 申请的 PKIBody 为 RevReqContent, 请求代码是 rr, 结构定义应符合 C.3.9。

CA 系统响应的 PKIBody 为 RevRepContent, 响应代码是 rp, 结构定义应符合 C.3.10。

6.2.2.3 证书更新协议

申请的 PKIBody 为 CertReqMessages, 请求代码是 kur, 结构定义应符合 C.3.5。通常情况下, 可为每一个更新的密钥提供 SubjectPublicKeyInfo, KeyId 和 Validity 字段。

响应的 PKIBody 为 CertRepMessage, 响应代码是 kup。该消息与证书申请的响应相同。

6.2.2.4 证书冻结协议

证书冻结应是证书临时撤销, 证书冻结请求与响应代码为 fr 和 fp, 结构定义应符合 C.3.20。

6.2.2.5 证书解冻协议

证书解冻应是证书冻结的反动作, 表示证书重新启用, 证书解冻请求与响应代码为 ufr 和 ufp, 结构定义应符合 C.3.21。

6.2.2.6 密钥恢复协议

PKI 可为用户提供密钥恢复的功能, 密钥恢复申请的 PKIBody 为 CertReqMessages, 请求代码是 krr, 结构定义应符合 C.3.7。

密钥恢复响应的 PKIBody 为 KeyRecRepContent, 响应代码是 krp, 结构定义应符合 C.3.8。对某些状态值(如 waiting), 可选字段都不出现。

6.3 CA 系统与 KM 系统间协议

6.3.1 协议概述

CA 系统与 KM 系统间的密钥服务包括密钥对申请、密钥对恢复和密钥对撤销, 每个步骤按照请求-响应的步骤执行。CA 系统与 KM 系统通信流程见图 4 所示。当 CA 系统在生成终端加密证书、更新加密证书或者撤销加密证书时, 首先组织密钥服务请求, 然后发送到 KM 系统, 并延缓自身的事务处理过程, 等待 KM 系统响应返回。KM 系统在接收到来自 CA 系统的请求后, 检查确定请求合法性, 处理服务请求, 并将结果返回给 CA 系统。



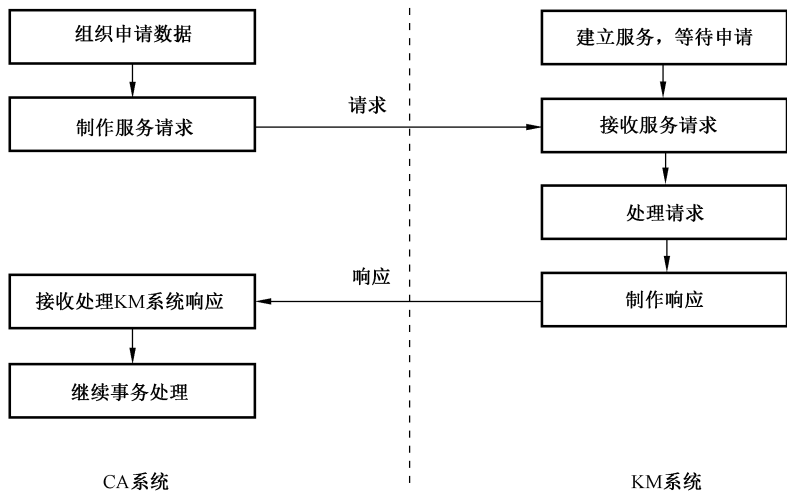


图 4 CA 系统与 KM 系统通信流程

协议内容如下。

- a) 请求,也称密钥服务请求,包含 CA 系统请求的类型、性质以及特性数据等,该请求将被发送到 KM 系统并得到服务响应。服务请求包括如下数据:
- 协议版本(当前版本为 2);
 - 服务请求标识符;
 - CA 标识符;
 - 扩展的请求信息;
 - 请求信息的签名。
- b) 响应,是 KM 系统对来自 CA 系统请求的处理响应。KM 系统的响应包括如下数据:
- 协议版本(当前版本为 2);
 - 响应标识符;
 - KM 标识符;
 - 响应信息;
 - 响应信息的签名。
- c) 异常情况,当 CA 系统和 KM 系统任何一方处理发生错误时,应向对方发送错误信息。错误可为:
- 验证请求失败:KM 系统验证来自 CA 系统证书或 CA 系统请求数据失败,CA 系统收到后应重新进行申请;
 - 内部处理失败:KM 系统处理 CA 系统请求过程中发生内部错误,通知 CA 系统该请求处理失败,需要重新申请。

采用抽象语法表示法(ASN.1)描述具体协议内容。如无特殊说明,默认为显式标记。

6.3.2 协议消息数据结构

6.3.2.1 密钥请求

6.3.2.1.1 请求数据格式

CA 系统请求的基本格式如下。

CAResponse ::= SEQUENCE{

```
ksRequest      KSRequest,
signatureAlgorithm AlgorithmIdentifier,
signatureValue  OCTETSTRING
}
```

其中:

```
KSRequest ::= SEQUENCE{
    version      Version DEFAULT v2,
    caName       EntName,
    requestList  SEQUENCE OF Request,
    requestTime  GeneralizedTime,
    taskNo       INTEGER
}
Version ::= INTEGER {v2(1)}
EntName ::= SEQUENCE {
    hashAlgorithm      AlgorithmIdentifier,
    entName            GeneralName,
    entPubKeyHash      OCTETSTRING,
    serialNumber       CertificateSerialNumber
}
CertificateSerialNumber ::= INTEGER
Request ::= CHOICE{
    applyKeyReq      [0]IMPLICIT ApplyKeyReq,
    restoreKeyReq    [1]IMPLICIT RestoreKeyReq,
    revokeKeyReq     [2]IMPLICIT RevokeKeyReq
}
```

6.3.2.1.2 KSRequest 及其结构解释

6.3.2.1.2.1 概述

KSRequest 结构包含了请求语法中的重要信息,具体包括版本、请求者标识符、请求类型、请求时间和任务序列号。

6.3.2.1.2.2 版本

version 版本描述了请求语法的版本号,当前版本为 2,取整型值 1。

6.3.2.1.2.3 请求者标识符

caName 请求者标识符是申请者的唯一名称,该值由 CA 系统和 KM 系统约定。entName 结构中,entPubKeyHash 是申请者公钥的杂凑值。该值将通过对发布者证书中的主体公钥字段(不含标记和长度)进行计算。hashAlgorithm 字段表示计算该杂凑值所使用的杂凑算法。serialNumber 是申请者的证书序列号。

6.3.2.1.2.4 请求类型

Request 请求包类型,值为 applyKey 时表明该包为申请密钥申请包,值为 restoreKey 时表明该包

为恢复密钥申请包,值为 revokeKey 时表明该包为撤销密钥申请包。

Request 的三种数据格式如下。

a) ApplyKeyReq 包。ApplyKeyReq 包为密钥申请格式包,其格式如下。

```

ApplyKeyReq ::= SEQUENCE {
    appKeyType      AlgType,
    appKeyLen       AppKeyLen,
    retAsymAlg      AlgType,
    retSymAlg       AlgType,
    retHashAlg      AlgType,
    appUserInfo     AppUserInfo
}
AlgType ::= AlgorithmIdentifier
AppKeyLen ::= INTEGER
AppUserInfo ::= SEQUENCE {
    userCertNo      CertificateSerialNumber,
    userPubKey      SubjectPublicKeyInfo,
    notBefore       GeneralizedTime,
    notAfter        GeneralizedTime,
    userName        [0]OCTETSTRING OPTIONAL,
    dsCode          [1]FreeText OPTIONAL,
    extendInfo      [2]FreeText OPTIONAL
}

```

密钥申请包的参数说明如下。

- AlgType:表明使用的非对称算法、对称算法、杂凑算法等算法类型。其中,appKeyType 为申请的加密密钥对的类型,retAsymAlg、retSymAlg、retHashAlg 分别为 KM 系统响应数据包中非对称算法、对称算法、杂凑算法类型。
- AppKeyLen:表示申请的密钥强度。如十进制 256 表示申请 256 比特的密钥。
- AppUserInfo:表示申请包中对应用户信息,依次为终端用户加密证书序列号、终端用户保护公钥、密钥有效起始时间、密钥截止时间、用户姓名、地区代码以及扩展信息。
- userCertNo:表示终端用户加密证书序列号。对于同一个 CA 系统,其申请时提交的用户加密证书序列号应是唯一的,如果需要使用原有证书序列号再次申请密钥对,则应当先请求将原申请密钥对撤销。
- userPubKey:表示终端用户保护公钥。该公钥应由终端用户证书载体生成,在本系统中用来在密钥传递中保护用户加密私钥。该公钥宜采用终端用户签名公钥。
- userName:标识用户名称。当 CA 系统将该项提交给 KM 系统时,KM 系统应当将其保存到库中。
- dsCode:标识地区代码。CA 系统可使用该项标识密钥的地区属性,当 dsCode 存在时,KM 系统应将其保存到库中。
- extendInfo:扩展信息。用来表示 CA 系统需要往 KM 系统发送关于该用户的个性信息。比如要求 KM 按照区域管理密钥,CA 系统可利用该域填写该用户的区域信息,KM 系统则可读出该域值后保存到库中。该域最大应能处理 100 字节数据。

b) RestoreKeyReq 包。RestoreKeyReq 包为密钥恢复格式包,其具体格式如下。

```

RestoreKeyReq ::= SEQUENCE {

```

```

    retAsymAlg    AlgType,
    retSymAlg     AlgType,
    retHashAlg    AlgType,
    userCertNo    CertificateSerialNumber,
    userPubKey    SubjectPublicKeyInfo
}

```

密钥恢复请求包的参数说明如下。

- AlgType: 表明使用的非对称算法、对称算法、杂凑算法类型。其中, retAsymAlg、retSymAlg、retHashAlg 分别为 KM 系统响应数据包中非对称算法、对称算法、杂凑算法类型。
- userCertNo: 为用户加密证书序列号。
- userPubKey: 为终端保护公钥。

c) RevokeKeyReq 包。RevokeKeyReq 包为密钥撤销格式包, 其具体格式如下。

```

RevokeKeyReq ::= SEQUENCE {
    userCertNo CertificateSerialNumber
}

```

UserCertNo 为用户证书序列号。

6.3.2.1.2.5 请求时间

requestTime 为请求生成时间, 该时间即为 CA 系统产生请求的时间。

6.3.2.1.2.6 任务序列号

taskNo 为请求任务序列号, 该任务序列号是申请者用来区分多次申请时候的一个标识符, 以确保 KM 系统和 CA 系统能正确关联请求-响应过程。KM 系统应能处理不大于 20 字节的任务序列号, 而 CA 系统应确保不使用大于 20 字节的任务序列号。

6.3.2.2 密钥响应

6.3.2.2.1 响应数据格式

KM 系统响应的格式如下:

```

KMRespond ::= SEQUENCE {
    ksRespond      KSRespond,
    signatureAlgorithm AlgorithmIdentifier,
    signatureValue  OCTET STRING
}

```

其中:

```

KSRespond ::= SEQUENCE {
    version          Version DEFAULT v2,
    KMName           EntName,
    respondList      SEQUENCE OF Respond,
    respondTime      GeneralizedTime,
    taskNo           INTEGER
}

```

```

Version ::= INTEGER {v2(1)}

```



```
Respond ::= CHOICE{
    applykeyRespond      [0]IMPLICIT RetKeyRespond,
    restorekeyRespond    [1]IMPLICIT RetKeyRespond,
    revokekeyRespond     [2]IMPLICIT RevokeKeyRespond,
    errorPkgRespond      [3]IMPLICIT ErrorPkgRespond
}
```

6.3.2.2.2 KSRespond 及其结构解释

6.3.2.2.2.1 概述

KSRespond 结构包含了响应语法中的重要信息,具体包括版本、响应者标识符、响应类型、响应时间和任务序列号。

6.3.2.2.2.2 版本

version 为响应语法的版本号,当前版本为 2,取整型值 1。

6.3.2.2.2.3 响应者标识符

KMName 为响应者标识符,其成员分别为响应者的名称、响应者公钥的杂凑值、杂凑算法以及响应者的证书序列号。

6.3.2.2.2.4 响应类型

Respond 为响应类型,当值为 applyRespond 时该包为申请密钥响应包,为 restoreRespond 时该包为恢复密钥响应包,值为 revokeRespond 时该包为撤销密钥响应包。

Respond 响应子包共有如下三种数据格式。

- a) retKeyRespond 包。retKeyRespond 包为密钥响应格式包,为处理申请密钥、恢复密钥申请时响应申请者的数据包,其具体格式如下。

```
retKeyRespond ::= SEQUENCE {
    userCertNo      CertificateSerialNumber,
    retPubKey       SubjectPublicKeyInfo,
    retPriKey       SM2EnvelopedKey
}
```

在密钥响应包中:

- UserCertNo:指用户加密证书序列号,该项从 CA 系统申请包中取值;
- retPubKey:指返回给申请者的用户加密公钥数据;
- retPriKey:指返回给申请者的用户加密私钥数据信封。

- b) RevokeKeyRespond 包。RevokeKeyRespond 包为密钥撤销响应格式包,为处理密钥撤销时响应申请者的数据包,其具体格式如下。

```
RevokeKeyRespond ::= SEQUENCE {
    userCertNo      CertificateSerialNumber
}
```

UserCertNo 指定用户加密证书序列号,该项值取自申请包。

- c) ErrorPkgRespond 包。ErrorPkgRespond 包为错误包,在处理密钥服务请求出错时,KM 系统使用本包响应申请者。其具体格式如下。

```
ErrorPkgRespond ::= SEQUENCE {  
    errNo          INTEGER,  
    errDesc        [0]FreeTextOPTIONAL  
}
```

UserCertNo 指定用户加密证书序列号,该项值取自申请包。

6.3.2.2.2.5 响应时间

respondTime 为响应生成时间,该时间即为 KM 系统产生响应的时间,采用 GeneralizedTime 语法表示。

6.3.2.2.2.6 任务序列号

taskNO 为响应的任务序列号,该任务序列号值取自申请者数据包。

6.4 CA 系统与资料库间协议

6.4.1 协议概述

CA 系统和资料库间的协议主要包括 CA 系统和 LDAP、OCSP 服务间的协议。CA 系统与资料库一般属于同一责任主体,二者之间的协议可能具有不同的实现方式,此处定义仅作为常用参考。

证书状态发布有两种方式,一种是 PkixIssue 方式,按照 CA 系统到 LDAP 服务的发布协议中发布到 OCSP 服务的规定。另一种是 PKIMessage 方式。

6.4.2 CA 系统与 LDAP 服务间协议

证书与证书撤销链发布是指 CA 系统把新签发的证书与证书撤销链送到 LDAP 目录服务端,以供用户查询、下载。

CA 系统到 LDAP 服务间的数据传输应符合 GB/T 25056 的规定。

CA 系统到 LDAP 服务的发布协议消息包结构为 PkixIssue(见 6.4.4),回应数据为 PkixIssueResponse(见 6.4.4)。如果 CA 系统收到回应后验证签名不通过或传输随机数不同,此次发布失败,下次应重新打包再发布。

6.4.3 CA 系统与 OCSP 服务间发布协议

当 CA 系统已经或将要作废一个特定的证书时,可发布一个有关该事件(可能是将要发生的事件)的告示。

CA 系统可使用这样的告示来警告(或通知)一个申请者其证书将要(或已经)作废。这一消息通常用于作废请求并不是由相关证书的 subject 发起的情况。

6.4.4 协议消息数据结构

CA 系统到 LDAP 服务的发布数据结构:

```
PkixIssue ::= SEQUENCE{  
    PkixIssueInfo          TBSISSUE,  
    --发布内容  
    SignatureAlgorithm      SignatureAlgorithmIdentifier,  
    CASignature              Signature  
}
```

Signature 域包含了对 PkixIssueInfo 域进行数字签名的结果,签名的结果按照 ASN.1 编码成 BIT-STRING 类型。

```

TBSISSUE ::= SEQUENCE{
    version                Version,
    --版本号,值为 1
    type                   INTEGER,
    --1:向 LDAP 发送证书,
    --2:向 LDAP 发送作废证书序列号,
    --3:向 LDAP 发送作废证书链,
    --4:向 OCSP 发送证书状态
    transNonce             OCTET STRING OPTIONAL,
    --包内随机数,长度最大不超过 20 字节
    number                 INTEGER OPTIONAL,
    --包内证书或证书状态或证书撤销链数目
    time                   Generalizedtime,
    --接收方比较此时间,根据约定时间延迟确定是否接收包内容
    cert                   [0] SEQUENCE SIZE (1..MAX) OF Certificate OPTIONAL,
    certstatue             [1] SEQUENCE OF SEQUENCE{
        certId             CertID,
        beforetime         Generalizedtime,
        endtime            Generalizedtime,
        statue             INTEGER,
        --0 有效,1 无效,无效时有原因项
        statuetime         Generalizedtime,
        statuereasonRode   CRLReason OPTIONAL
    } OPTIONAL,
    mCRL                   [2] SEQUENCE OF SEQUENCE{
        stateOrProvinceNum INTEGER,
        --区域区别号
        CRLSegment         INTEGER,
        CRLnumber          INTEGER,
        --证书链基本序列号
        DeltareCRLnumber   INTEGER OPTIONAL,
        --证书链增量序列号
        certificateList     SEQUENCE{
            --作废证书链
            tbscertList     TBScertList,
            SignatureAlgorithm AlgorithmIdentifier,
            SignatureValue  BITSTRING
        }
    } OPTIONAL,
    signatureAlgorithm     AlgorithmIdentifier OPTIONAL,
    --签名算法

```

```

signatureValue          BITSTRING          OPTIONAL,
--CA 系统的签名
}

```

CRLSegment 是证书链的块号。为防证书链太大,一般按证书号分块,同块内证书在同一条链中。CRLReason 的定义应符合 GB/T 20518—2018 的 5.3.4.7.1。

LDAP 服务到 CA 系统的回应包结构:

```

PkixIssueResponse ::= SEQUENCE {
    PkixIssueResponseInfo TBSISSUERResponse,
    SignatureAlgorithm     SignatureAlgorithmIdentifier,
    ResponseSignature       Signature
}
TBSISSUERResponse SEQUENCE ::= {
    version                Version, --版本号, 目前为 1
    type                   INTEGER,
                           1: 向 LDAP 发送证书,
                           2: 向 LDAP 发送作废证书序列号,
                           3: 向 LDAP 发送作废证书链,
                           4: 向 OCSP 发送证书状态
    transNonce             OCTETSTRING     OPTIONAL,
    --包内随机数, 长度最大不超过 20 字节
    number                 INTEGER         OPTIONAL,
    --包内证书或证书状态或证书撤销链数目
    time                   Generalizedtime,
    --接收方时间
    Responsestatue         INTEGER
    --0 为正常接收; 1 为未接收, 请下次重发
}
Version ::= INTEGER

```

CA 系统与 OCSP 服务间 PKIMessage 方式的发布协议数据结构应符合 C.3.12。

6.5 终端与资料库间协议

6.5.1 协议概述

终端与资料库间的协议包括 LDAP、OCSP 方式。

- a) LDAP 方式。LDAP 目录可用来存储多种信息,作为 PKI 的核心组件,其作用主要用来存放证书与 CRL,供用户查询。
- b) OCSP 方式。OCSP 作为查询 CRL 的替代方法或补充方法,实时获得数字证书撤销状态相关信息时,其是不可或缺的。OCSP 能使应用程序获得某个目标证书的撤销状态,OCSP 可提供比检查 CRL 更实时的撤销状态信息,还可提供附加的状态信息。OCSP 客户端向 OCSP 响应器发出一个状态请求时,应暂停接受待验证的证书,直到响应器提供响应为止。

6.5.2 终端与 LDAP 服务间协议

6.5.2.1 LDAP 消息数据结构

LDAP 消息协议数据单元 LDAPMessage 如下。

LDAPMessage ::= SEQUENCE {
 messageID MessageID,
 protocolOp CHOICE {
 bindRequest BindRequest,
 bindResponse BindResponse,
 unbindRequest UnbindRequest,
 searchRequest SearchRequest,
 searchResEntry SearchResultEntry,
 searchResDone SearchResultDone,
 searchResRef SearchResultReference,
 modifyRequest ModifyRequest,
 modifyResponse ModifyResponse,
 addRequest AddRequest,
 addResponse AddResponse,
 delRequest DelRequest,
 delResponse DelResponse,
 modDNRequest ModifyDNRequest,
 modDNResponse ModifyDNResponse,
 compareRequest CompareRequest,
 compareResponse CompareResponse,
 abandonRequest AbandonRequest,
 extendedReq ExtendedRequest,
 extendedResp ExtendedResponse,
 ...,
 intermediateResponse IntermediateResponse },
 controls[0]Controls OPTIONAL }

MessageID ::= INTEGER (0 ..maxInt)
maxInt INTEGER ::= 2147483647 --(2³¹-1)--

LDAP 消息(LDAPMessage)的功能是给所有的协议交换定义一个包含通用字段的封装。

除了在上面定义的 LDAPMessage,在定义协议操作时,也使用下面这些定义。

LDAPString ::= OCTET STRING -- UTF-8 编码,[ISO 10646]字符

LDAPString 是一种符号上的方便表示,其表明尽管 LDAPString 是一种用字符串类型来编码的串,但实际上该串能使用的合法字符集由 IA5 字符集限定。

LDAPDN ::= LDAPString

RelativeLDAPDN ::= LDAPString

一个 LDAPDN 和一个 RelativeLDAPDN 被独立地定义,分别代表一个标识名和一个相对标识名。

AttributeDescription ::= LDAPString

--被约束到 RFC 4512 的<attributedescription>

属性描述的定义和编码规则见 RFC 4512 的 2.5。

AttributeValue ::= OCTETSTRING

属性值定义对应具体的上下文见 RFC 4517 和其他文档。

AttributeValueAssertion ::= SEQUENCE {

attributeDesc AttributeDescription,

assertionValue AssertionValue

}

AttributeValueAssertion 的类型定义与 X.500 目录标准类似,包含一个属性描述和匹配规则的声称值。

AssertionValue ::= OCTET STRING

AssertionValue 的内容取决于 LDAP 操作的上下文。

LDAPResult ::= SEQUENCE {

resultCode ENUMERATED {

Success (0),

operationsError (1),

protocolError (2),

timeLimitExceeded (3),

sizeLimitExceeded (4),

compareFalse (5),

compareTrue (6),

authMethodNotSupported (7),

strongAuthRequired (8),

--9reserved--

referral (10),

adminLimitExceeded (11),

unavailableCriticalExtension (12),

confidentialityRequired (13),

saslBindInProgress (14),

noSuchAttribute (16),

undefinedAttributeType (17),

inappropriateMatching (18),

constraintViolation (19),

attributeOrValueExists (20),

invalidAttributeSyntax (21),

--22-31 unused--

noSuchObject (32),

aliasProblem (33),

invalidDNSyntax (34),

--35 reserved for undefined isLeaf--

aliasDereferencingProblem (36),

--37-47 unused--

inappropriateAuthentication (48),

invalidCredentials (49),

```

insufficientAccessRights      (50),
busy                          (51),
unavailable                    (52),
unwillingToPerform            (53),
loopDetect                    (54),
--55-63unused--
namingViolation               (64),
bjectClassViolation           (65),
notAllowedOnNonLeaf           (66),
notAllowedOnRDN               (67),
entryAlreadyExists            (68),
objectClassModsProhibited     (69),
--70 reserved for CLDAP--
affectsMultipleDSAs           (71),
--72-79unused--
other                         (80)
...},
matchedDN      LDAPDN,
errorMessage   LDAPString,
referral       [3] Referral OPTIONAL
}

```

LDAPResult 是从服务端返回给客户端的结构,用以指明操作成功或失败的结构。对不同的请求,服务端返回包含 LDAPResult 结构中的不同域的回应给客户端,来指明协议操作请求的最终状态。具体含义见 RFC 4511 的附录 A,对应枚举值的扩展定义见 RFC 4520 的 3.8。

对于服务端,此结构中的 errorMessage 域应用来返回一个包含可读文本的错误诊断的 ASCII 串给客户端。由于这个错误诊断不是标准的,在实现中不应依赖这些返回值。如果服务端不返回一个文本的错误诊断,LDAPResult 结构中的 errorMessage 应包含一个长度为零的字符串。

若 resultCode 为 noSuchObject、aliasProblem、invalidDNsyntax 及 aliasDereferencingProblem,相应的 matchedDN 域应设为在 DIT 中最为匹配的条目,而且应是所提供名字的简短格式。或者,如果别名已被丢弃,应返回结果名字。在其他情况下,matchedDN 域都应设为 NULLDN(即长度为零的字符串)。

客户端和服务端之间的协议内容,包括绑定、解除绑定、查询、插入、修改、删除、丢弃等。依据规定,用户只能访问从 LDAP,所以 LDAP 应拒绝用户的插入、修改、删除功能、协议中通用部分说明如下。

绑定操作的功能是在客户端和服务端之间进行初始化,并允许服务端对客户端进行认证。绑定请求定义如下。

```

BindRequest ::= [APPLICATION 0]SEQUENCE {
    version      INTEGER (1 ..127),
    name          LDAPDN,
    Authentication AuthenticationChoice
}
AuthenticationChoice ::= CHOICE {
    simple      [0] OCTETSTRING,--1 和 2 保留

```

```

        sasl          [3] SaslCredentials
        ...
    }
    SaslCredentials ::= SEQUENCE {
        mechanism      LDAPString,
        Credentials     OCTET STRING OPTIONAL
    }

```

绑定请求的参数说明如下。

- 版本：一个版本号指定本协议所使用的版本。本文档描述的是 LDAP 版本 3。由于没有版本选择谈判，客户端把此参数设为它所希望的值；如果服务端不支持对应的版本，应返回结果码是“protocolError”的 BindResponse。
- 名称：客户端希望绑定的目录对象的名称。这个域可是空值（一个长度为零的字符串），用来表示匿名绑定或适用 SASL 认证时。
- 认证：认证用的信息。认证类型的扩展定义见 RFC 4520 的 3.7。如果服务端不支持客户端提供的选项，应返回结果码是“authMethodNotSupported”的 BindResponse。

绑定操作需要如下定义的绑定回应。

```

BindResponse ::= [APPLICATION 1] SEQUENCE {
    ldapResult          COMPONENTS OF LDAPResult,
    serverSaslCreds     [7] OCTET STRING OPTIONAL
}

```

放弃绑定操作的功能是临时终止一个协议会话，放弃绑定操作定义如下。

```
UnbindRequest ::= [APPLICATION 2] NULL
```

在收到一个客户端的放弃绑定请求后，服务端可假设发送请求的客户端已终止该会话，所有已收到的还未处理的请求可被丢弃。

丢弃操作的功能是允许客户端请求服务端终止一个已发出未完成的操作请求。丢弃操作定义如下。

```
AbandonRequest ::= [APPLICATION 16] MessageID
```

在丢弃操作中并没有定义相应的回应。在发送一个丢弃操作后，一个客户端可能期望丢弃请求中包含的消息 ID 标识的操作已被丢弃。如果服务端在把一个查找操作的回应发送给客户端时，收到对该查找操作的丢弃请求，服务端应停止发送回应，并立即终止该查找操作。

Abandon、Bind、Unbind 和 StartTLS 操作不能被丢弃。

6.5.2.2 证书查询与下载协议

查找操作允许客户端请求代表他的服务端进行一次查找。查找请求定义如下。

```

SearchRequest ::= [APPLICATION 3] SEQUENCE {
    baseObject    LDAPDN,
    scope         ENUMERATED {
        baseObject          (0),
        singleLevel         (1),
        wholeSubtree        (2),
        ... },
    derefAliases  ENUMERATED {
        neverDerefAliases   (0),

```



```

        derefInSearching          (1),
        derefFindingBaseObj      (2),
        derefAlways              (3)
    },
    sizeLimit      INTEGER (0 ..maxInt),
    timeLimit      INTEGER (0 ..maxInt),
    attrsOnly      BOOLEAN,
    filter          Filter,
    attributes      AttributeSelection
}
AttributeSelection ::= SEQUENCE OF selector LDAPString
Filter ::= CHOICE {
    and          [0]SET SIZE (1..MAX) OF filter Filter,
    or           [1]SET SIZE (1..MAX) OF filter Filter,
    not          [2]Filter,
    equalityMatch [3]AttributeValueAssertion,
    substrings   [4]SubstringFilter,
    greaterOrEqual [5]AttributeValueAssertion,
    lessOrEqual  [6]AttributeValueAssertion,
    present      [7]AttributeDescription,
    approxMatch  [8]AttributeValueAssertion,
    extensibleMatch[9]MatchingRuleAssertion,
    ...}
SubstringFilter ::= SEQUENCE {
    type          AttributeDescription,
    substrings     SEQUENCE SIZE (1..MAX) OF substring CHOICE {
        initial    [0]AssertionValue,
        any        [1]AssertionValue,
        final      [2]AssertionValue,
    }
}
MatchingRuleAssertion ::= SEQUENCE {
    matchingRule  [1]MatchingRuleId OPTIONAL,
    type          [2]AttributeDescription OPTIONAL,
    matchValue    [3]AssertionValue,
    dnAttributes  [4]BOOLEAN DEFAULT FALSE
}

```

查找请求的参数说明如下。

- baseObject: 一个相对于要进行查找操作的基本对象条目。
- scope: 指示要查找的范围。该域可能的语义上的值与目录查找操作中的范围域的语义值是一致的。
- derefAliases: 指示在操作中别名对象该如何处理。该域可能的语义上的值按照升序的顺序进行。

- neverDerefAliases:在查找或定位查找的基本对象时不丢弃别名引用。
- derefInSearching:在查找过程中,丢弃基本对象的下一级的别名引用,但对基本对象进行定位时,不丢弃别名引用。
- derefFindingBaseObject:在定位基本对象时,丢弃别名引用,但在查找基本对象的下一级时,不丢弃别名引用。
- derefAlways:在定位基本对象和查找基本对象的下一级时都丢弃别名引用。
- sizeLimit:限制了可返回的最大查找结果的条目数量。如果该域的值为0,表示不限制查找的sizelimit。
- timeLimit:限制了一个查找最多允许的时间(以秒计算)。如果该域的值为0,表示不限制查找的timelimit。
- attrsOnly:指示在返回的查找结果中是否应包含属性类型和属性值,或者仅包含属性类型。如果该域设为TRUE,仅返回属性类型(没有值),如果该域设为FALSE,同时返回属性类型和属性值。
- filter:一个过滤器定义了一个应满足的条件,以使该查找能匹配给定的条目。
- attributes:应返回的查找结果中的每一个条目的属性列表。一个空的列表表示查找结果应包含每一个条目的所有属性。

查找操作的结果包括0或更多的 SearchResultEntry 和/或 SearchResultReference 消息,并跟随单独的一条 SearchResultDone 消息。

```
SearchResultEntry ::= [APPLICATION 4] SEQUENCE {
    objectName      LDAPDN,
    attributes      PartialAttributeList
}
```

```
PartialAttributeList ::= SEQUENCE OF SEQUENCE {
    type            AttributeDescription,
    vals            SET OF AttributeValue
}
```

```
SearchResultReference ::= [APPLICATION 19] SEQUENCE SIZE (1..MAX) OF uri URI
```

```
SearchResultDone ::= [APPLICATION 5] LDAPResult
```

每一个 SearchResultEntry 代表查找操作找到一个条目,每一个 SearchResultReference 代表查找操作还有一个区域没有探索。SearchResultEntry 和 SearchResultReference 消息可为任意顺序,随后服务端返回包含成功或失败细节指示的 SearchResultDone 响应。

6.5.2.3 CRL 查询与下载协议

证书或证书撤销列表是以一条条数据的形式存放在 LDAP 不同的分支下。LDAP 根据“baseObject”与属性值查询满足条件的记录返回用户查询下载的证书或 CRL。

baseObject 是查找的基本条件,具体查找一条 CRL 应根据目录树的设计区别对待。CRL 目录有两种设计。

- a) 根据证书号分段,如 201 段 CRLLDAPDN 存放“serialnumber=???,ou=crl201,dc=isc,dc=com”,202 段 CRLLDAPDN 存放“serialnumber=???,ou=crl202,dc=isc,dc=com”。其中 CRLNumber 放到 serialnumber 中,分段号组合到 ou 中,具体的区别名放到 dc 中。
- b) 增量 CRLCRLLDAPDN 存放“serialnumber=???,ou=dcrl,dc=isc,dc=com”? 因为增量 CRL 的序列号是 CRL 基本序列号的累加,如基本号为 1,增量号为 2……9,下一个基本号为 10,接下来的增量号为 11……19,这个递增的号码不应重复,把这个递增的号码放入 serial-

number。

对于上述 a) 的情况,若查找 CRLNumber 为 100、分段号为 10 的 CRL,baseObject 可这样组织:“serialnumber=100,ou=crl10,dc=isc,dc=com”,发送这样的请求就可获得具体的 CRL 列表。

对于上述 b) 的情况,若查找 serialnumber 为 100,baseObject 可这样组织:“serialnumber=100,ou=dcrl,dc=isc,dc=com”,发送这样的请求就可获得具体的 CRL 列表。对于一个应用来说,如果这个列表是基本表,时间也满足要求,通过这个表就可知道证书是否作废;如果这个列表是增量表,时间也满足要求,通过这个表还不能知道证书是否作废,这时还应下载对应此表的基本表以及到目前表为止的增量表,经过每个表的检查方知具体证书的作废情况。

6.5.3 终端与 OCSP 服务间协议

应符合 GB/T 19713。



附 录 A
(规范性)
必选的证书管理消息结构

A.1 概述

必选的证书管理消息结构包含了终端与 PKI 管理组件间基于 PKI 消息进行证书管理实现时应支持的 PKI 消息详细描述,涉及的 PKIMessage 类型包括:

- 初始的注册/认证;
- 基本认证方案;
- 证书请求;
- 密钥更新。

A.2 消息结构解释的通用规则

消息结构解释的通用规则如下。

- a) 如果消息结构中没有可选(OPTIONAL)或缺省(DEFAULT)字段,相关的消息中也不能有相关字段(例如:接收者可以语法错误为由丢弃包含相关字段的消息)。如果强制字段具有较明显的值,则文中不应特别提及(例如:pvno 总是为 103)。
- b) 当结构中不止一个消息时,应分别进行描述。
- c) PKIMessage 结构的 algorithmIdentifiers 应分别描述。
- d) 一种特殊的 X.500 DN 称为“NULL-DN”;表示 DN 包含长度为 0 的 SEQUENCE OF RelativeDistinguishedNames(其 DER 编码为‘3000’H)。
- e) 如果某字段需要 GeneralName,但无合适的值(例如:终端在知道其名字之前产生了一个请求),则 GeneralName 为 X.500 NULL-DN(例如:CHOICE 的 Name 字段含有 NULL-DN)。此特殊值也称为“NULL-GeneralName”。
- f) 如果 GeneralName 未指定值,则相关 PKIMessage 字段的值为 NULL-GeneralName。此情况通常发生在某些消息 PKIHeader 的 sender 字段。
- g) 如果字段命名产生歧义,则使用“点”号来进行命名(例如:“certTemplate.subject”表示 subject 字段在 certTemplate 字段中)。
- h) “SEQUENCE OF types”作为消息的一部分时,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 A.5~A.7 中所有的 PKI 消息交换可由发送方发送 certConf 消息,同时响应方也可发送 PKIConfirm 消息。对于 protectionAlg 可使用任何认证方法(例如:基于口令的 MAC 或者签名)。

A.3 算法使用参数

表 A.1 包含了在证书管理协议中使用的算法定义。

表 A.1 证书管理协议使用的算法

Name:消息结构使用的标识符	Use:算法使用的原因和场景	算法取值
MSG_SIG_ALG	用签名保护 PKI 消息	基于 SM2 算法和 SM3 算法的签名,对象标识符 OID 取值为 1.2.156.10197.1.501,应符合 GB/T 33560
MSG_MAC_ALG	用 MAC 保护 PKI 消息	SM3_HMAC、SM4_MAC,对象标识符 OID 取值分别为 1.2.156.10197.1.401.2、1.2.156.10197.1.104.5
PROT_ENC_ALG	用于加密 PKIMessages 中传输的对称密钥(此对称密钥用于加密非对称算法私钥)的非对称算法	SM2,对象标识符 OID 取值 1.2.156.10197.1.301.3,应符合 GB/T 33560
PROT_SYM_ALG	用于加密私钥的对称加密算法(该对称算法的对称密钥使用 PROT_ENC_ALG 进行加密)	SM4,对象标识符 OID 取值 1.2.156.10197.1.104,应符合 GB/T 33560

A.4 所有权证明消息结构

当需要证明拥有所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)。表 A.2 为证明拥有请求证书中与验证公钥相应的签名私钥时,POP 字段中的字段、值及其说明信息。

表 A.2 签名 POP 字段中的字段、值及其说明

字段	值	说明
algorithmIdentifier	MSG_SIG_ALG	此证明只允许采用签名保护
signature	present	使用 MSG_SIG_ALG 计算的比特

在认证请求协议中,不是每一个 CA/RA 系统均需要所有权证明。如何进行 POP 是一个策略问题,每一个 CA 系统都在公布的 Policy OID 和 Certification Practice Statement 中进行明确表示。要求 CA/RA 系统把 POP 作为认证过程的一部分。所有终端均应能提供 POP,PKIX-CMP 协议的组件应支持 POP。

A.5 初始的注册/认证(基本认证方案)

(未初始化的)终端向 PKI 管理组件请求(第一个)证书。当 PKI 管理组件返回包含有证书的消息时,终端可发送证书确认。PKI 管理组件也可再返回 PKIConfirm,关闭交易。交互的所有消息均进行认证。

终端申请签名证书时,由本地产生密钥对,将私钥妥善保管,公钥发送给 PKI 管理组件,PKI 管理组件为其签发证书。终端申请加密证书时,由 CA 按照相关法规与标准要求为其产生密钥对并签发证书。

请求认证仅用于一个本地产生的公钥;对于更多的公钥,应使用独立的 PKIMessage。

终端应支持对与本地产生公钥相关联的私钥的所有权证明。

描述一种常见的实现方式,交互双方可通过实现方式的协商来实现互联互通。

前提条件包括：

终端可验证 PKI 管理组件的签名；

终端和 PKI 管理组件共享一个对称的 MAC 密钥(即,用于进行 MAC 计算的对称密钥)。

消息流：

步骤	终端		PKI 管理组件
1	format ir		
2		—>	ir —>
3			handle ir
4			format ip
5		<—	ip <—
6	handle ip		
7	format certConf		
8		—>	certConf(可选的)—>
9			handle certConf
10			format PKIConf
11		<—	PKIConf(可选的) <—
12	handle PKIConf		

在本消息结构中,终端在一个 PKIMessage 中包含所有(一个或者两个)CertReqMsg,同时 PKI 管理组件产生一个包含完整响应的 PKIMessage(如果请求了集中密钥生成,则包括可选的第二个密钥对)。

终端与 PKI 管理组件提前建立了共享密钥、referenceNumber 参考号码,也可与 PKI 管理组件提前确定 sender 和用于 CertTemplate 结构体 subject 的可识别名。宜共享秘密至少长 12 个字符。

ir:

字段	值
recipient	PKI name
--被要求产生证书的 PKI 管理组件的名字	
protectionAlg	MSG_MAC_ALG
--此请求只允许基于初始的认证密钥进行 MAC 保护	
senderKID	referenceNum
--索引号,由 PKI 管理组件预先发给终端(同时还有 MAC 密钥)	
transactionID	present
--特定的值,如果该值在 PKI 管理组件中已使用,PKI 管理组件应产生拒绝消息	
senderNonce	present
--128 (伪)随机比特	
freeText	any valid value
body	ir (CertReqMessages)
--仅支持一个或两个 CertReqMsg,若请求中只有一个 CertReqMsg,则响应只返回单证书	
--双证书(一个签名证书和一个加密证书)的申请须使用一个 PKIMessage 中的两个 CertReqMsg	

--如果请求更多的证书,请求须打包在不同的 PKIMessage 中	
CertReqMsg	one or two present
--细节如下, crm[0]表示第一个(应存在)	
--crm[1]表示第二个(可选的,且用于请求集中产生密钥)	
crm[0].certReq.certReqId	fixed value of zero
--消息中模板的索引	
crm[0].certReq.certTemplate	present
--应包含主体公钥值	
crm[0].pop. POPOSigningKey	optionally present if public key from crm[0].certReq.certTemplate is a signing key
--在交互中可能要求所有权证明 POP (见 A.4)	
crm[0].certReq.controls.archiveOptions	optionally present
--终端可请求归档本地产生的私钥	
crm[0].certReq.controls.publicationInfo	optionally present
--终端可要求公布证书	
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]	
--如果支持集中产生密钥,可用此短期的非对称加密密钥	
--(由终端产生)来加密(用对称密钥来加密)PKI 管理组件为终端产生的私钥	
crm[1].certReq.controls.archiveOptions	optionally present
crm[1].certReq.controls.publicationInfo	optionally present
protection	present
--使用 MSG_MAC_ALG 计算的比特	
ip:	
字段	值
sender	PKI name
--产生消息的 PKI 管理组件的名字	
messageTime	present
--PKI 管理组件产生消息的时间	
protectionAlg	MS_MAC_ALG
--这个本响应只允许 MAC 保护	
senderKID	referenceNum
--预先发给终端索引号	
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 管理组件对一个或者两个请求的响应	
--crc[0]表示第一个(应存在);crc[1]表示	
--第二个(仅存在于 ir 消息包含两个请求并且支持集中密钥产生)	
--双证书的响应应在一个 PKIMessage 中的 CertRepMessage 中完成	
crc[0].certReqId	fixed value of zero
--包含对相应 ir 消息中第一个请求的响应	
crc[0].status. status	present, positive values allowed:
status	"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	
crc[1].certReqId	fixed value of one
--应包含对相应 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
publicationInfo	optionally present
--表明证书已经发布	
protection	present

--使用 MSG_MAC_ALG 计算的比特	
extraCerts	optionally present
--PKI 管理组件可为终端提供附加的其他证书	
certConf:	
字段	值
sender	present
--与 ir 中的相同	
recipient	PKI name
--被要求产生证书的 PKI 管理组件的名字	
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
--此消息只允许使用 MAC 算法保护。MAC 是	
--基于共享的初始认证密钥实现的	
senderKID	referenceNum
--预先发给终端的索引号	
body	certConf
--符合 C.3.15 中 certConf 字段内容	
--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 计算的比特	
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
--此消息只允许 MAC 保护	
senderKID	referenceNum
body	PKIConf
protection	present

--使用 MSG_MAC_ALG 计算的比特

A.6 证书请求

(已经初始化的)终端向 PKI 管理组件请求证书。当收到包含证书的响应消息时,终端可发送证书确认。PKI 管理组件可再发回 PKIConfirm,关闭交易。其中所有消息均进行认证。

本消息结构中的消息交互流程与 A.5 有以下不同:

- 发送者名字应存在;
- 在 request、response、certConfirm 和 PKIConfirm 消息中应支持 MSG_SIG_ALG 的 protectionAlg(也支持 MSG_MAC_ALG);
- senderKID 和 recipKID 仅在请求消息验证时才存在;
- body 为 cr 或 cp;
- body 可能包含一个或两个 CertReqMsg 结构;
- 保护比特根据 protectionAlg 字段来计算。

A.7 密钥更新请求

(已经初始化的)终端向 PKI 管理组件请求证书(用于更新密钥对和/或已经拥有的相应证书)。收到包含证书的响应消息时,终端可发送证书确认。PKI 管理组件可再发回 PKIConfirm,关闭交易。其中所有消息均进行认证。

本消息结构中的消息交互流程与 A.5 有以下不同:

- 发送者名字应存在;
- 在 request、response、certConfirm 和 PKIConfirm 消息中应支持 MSG_SIG_ALG 的 protectionAlg(也支持 MSG_MAC_ALG);
- senderKID 和 recipKID 仅在请求消息验证时才存在;
- body 为 kur 或 kup;
- body 可能包含一个或两个 CertReqMsg 结构;
- 保护比特根据 protectionAlg 字段来计算。



附录 B
(资料性)
可选的证书管理消息结构

B.1 概述

可选的证书管理消息结构包含了终端与 PKI 管理组件间基于 PKI 消息进行证书管理实现时可支持的 PKI 消息详细描述,涉及的 PKIMessage 类型包括:

- 信息请求/响应;
- 使用外部身份证书的初始化;
- 撤销请求;
- 证书发布;
- 轮询请求/响应;
- 证书/PKI 确认内容;
- 证书冻结与解冻请求/响应;
- CRL 发布。



B.2 结构解释的通用规则

见 A.2。

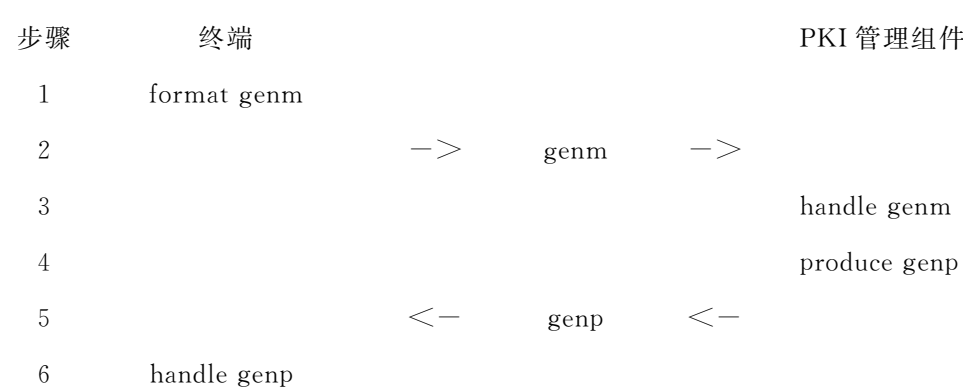
B.3 算法使用参数

见 A.3。

B.4 PKI 信息请求/响应

终端发送通用消息给 PKI 管理组件,以请求随后的证书管理操作中所需要的细节。RA/CA 系统用通用消息的响应消息来响应。如果 RA 系统产生响应,将简单地将从 CA 系统收到的消息向前转发,并可能在 PKIMessage 的 extraCerts 字段添加证书。本消息不需要从终端处获得确认消息。

消息流:



genM:

字段	值
recipient	PKI name

--证书中 issuerAltName 扩展或 issuer 字段包含的颁发者主题名称

protectionAlg	MSG_MAC_ALG or MSG_SIG_ALG
--任何一种鉴别保护算法	
SenderKID	present if required
--需要验证消息保护时需要存在	
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 计算的比特	
genP:	
字段	值
sender	PKI name
--产生消息的 PKI 管理组件的名字	
protectionAlg	MSG_MAC_ALG or MSG_SIG_ALG
--任何一种鉴别保护算法	
senderKID	present if required
--需要验证消息保护时需要存在	
body	genp (GenRepContent)
CAProtEncCert	present (object identifier one of PROT_ENC_ALG), with relevant value
--终端发送加密信息时使用 (例如, 用于私钥恢复)	
SignKeyPairTypes	present, with relevant value
--为主体公钥进行认证的签名算法标识符集合	
EncKeyPairTypes	present, with relevant value
--为主体公钥进行认证的加密算法标识符集合	
PreferredSymmAlg	present (object identifier one of PROT_SYM_ALG), with relevant value
--在随后的 PKI 消息中使用的对称加密算法	
CAKeyUpdateInfo	optionally present, with relevant value
--关于相关根 CA 密钥对的信息	
CurrentCRL	optionally present, with relevant value
--完整的 CRL 副本	
protection	present
--使用 MSG_MAC_ALG 或 MSG_SIG_ALG 计算的比特	
extraCerts	optionally present
--用于给终端发送证书	
--RA 系统可在此处添加其证书	

B.5 使用外部身份证书进行初始化

(未初始化的)终端希望初始化为 PKI 组件(其 CA 系统为 CA-1)的一员。其使用一个原来已经存在的由另一个(外部)CA 系统, CA-X 签发的身份证书进行身份鉴别。这要求 CA-1 与 CA-X 已经建立信任关系, 以便 CA-1 能验证由 CA-X 签发的终端的身份证书。另外, 在终端的个人安全环境(PSE)中

建立一些机制,允许其可鉴别和验证由 CA-1 签名的 PKIMessage(例如,PSE 包含为 CA-1 的公钥签发的证书,此证书是由终端信任的另一个 CA 系统签发)。

终端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,终端可返回证书确认。CA-1 可回应 PKIConfirm 来结束交易。所有消息都经过签名(终端发送的消息使用与外部身份证书中的公钥相对应的私钥签名;CA-1 发送的消息用与终端的 PSE 中信任的另一个证书公钥相对应的私钥签名)。

消息交互流程与 A.5 有以下不同:

- 终端和 CA-1 不共享对称 MAC 密钥(这些组件之间没有共享的秘密信息);
- ir 中发送者名字存在(且与外部身份证书中主体名一样);
- 所有消息均使用 MSG_SIG_ALG 的 protectionAlg;
- 在 ir 的 extraCerts 字段中携带外部身份证书;
- 不使用 senderKID 和 recipKID;
- body 为 ir 或 ip。



附 录 C
(规范性)
PKI 消息数据结构

C.1 PKI 消息综述

C.1.1 PKI 消息

PKI 消息是证书管理消息中的一部分,其结构如下所示。

```
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

在 PKIMessage 中:

- PKIHeader:包含 PKI 消息通用的信息;
- PKIBody:包含与具体类型的 PKI 消息相关的信息;
- PKIProtection:包含对 PKI 消息进行保护的比特串;
- extraCerts:包含对接收者可能有用的证书。

PKI 消息中所用到的公共数据结构见 C.2,特定操作数据结构见 C.3。

C.1.2 PKI 消息头

C.1.2.1 PKI 消息头结构

PKI 消息需使用消息头的某些信息进行寻址和交易识别。PKI 消息头携带于 PKI 消息的传输信息中,与 PKI 消息受到同等的保护。PKI 消息头的数据结构如下所示:

```
PKIHeader ::= SEQUENCE {  
    pvno                INTEGER{cmp1999(1),cmp2000(2),cmp2021(3),cmp2024(103)},  
    sender              GeneralName,  
    --标识发送者  
    recipient          GeneralName,  
    --标识预期的接收者  
    messageTime        [0] GeneralizedTime                OPTIONAL,  
    --产生此消息的时间  
    protectionAlg       [1] AlgorithmIdentifier            OPTIONAL,  
    --用于计算 protection 比特串的算法  
    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
--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[RFC 3629]编码的文本

在 PKIHeader 中参数说明如下。

——pvno: 指示协议版本。对于该版本 pvno 字段取固定值 103, 对于 GB/T 19714—2005 版本 pvno 字段取固定值 2。其他取值对应版本是 IETF RFC 标准版本(对于 RFC 2510 版本取固定值 1, 对于 RFC 9480 版本取固定值 3), 相关协议和消息定义不在描述范围内。关于 GB/T 19714—2005 和 GB/T 19714—202× 两个版本的协商处理问题见附录 D。

——sender: 包含 PKIMessage 发送者的名字。sender 字段与 senderKID 字段一起应能标识对该消息的保护进行验证所需要的密钥。如果发送方不清楚其相关信息[例如: 在初始化请求消息中, 终端不清楚自己的 DN(Distinguished Name, 唯一标识名)、电子邮件、IP 地址等], 则“sender”字段应包含一个“NULL”值, 即相对唯一标识的长度为 0。在此情况下, senderKID 字段应包含能向接收者指示验证消息所用的共享密钥信息的标识符。

——recipient: 包含 PKIMessage 接收者的名字。recipient 字段与 recipKID 字段一起应可用于验证对消息的保护。

——protectionAlg: 指示保护消息所使用的算法。

——senderKID 及 recipKID: 标识用于保护消息的密钥(recipKID 通常仅在使用 Diffie-Hellman 密钥保护消息时才要求)。如果需要唯一标识密钥(例如, 如果多个密钥与该发送者名字关联), 则应使用这些字段, 否则应省略这些字段。

——transactionID: 用于使消息的接收者将此消息与正在进行的交易相关联, 对于由多个请求/响应对组成的交易此字段是必选的。对于由单个请求/响应对组成的交易, 规则如下:

- 客户端可填充请求的 transactionID 字段;
- 如果服务端收到设置了 transactionID 字段的请求, 则其应将响应的 transactionID 字段设置为相同的值;
- 如果服务端接收到缺少 transactionID 字段的此类请求, 则其可选择设置响应的 transactionID 字段。

对于由多个请求/响应对组成的交易, 规则如下:

- 客户端应为第一个请求生成 transactionID;
- 如果服务端收到设置了 transactionID 字段的请求, 则其应将响应的 transactionID 字段设置为相同的值;
- 如果服务端接收到缺少 transactionID 字段的请求, 则其应使用服务端生成的 ID 填充响应的 transactionID 字段;
- 后续请求和响应都应应将 transactionID 字段设置为此值。

在使用 transactionID 的情况下, 给定客户端不应(同特定服务端)有多个正在进行的具有相同

transactionID 的交易。服务端可自由要求 transactionID 的唯一性。如果服务端不能区分客户端,则其应要求 {client, transactionID} 元组或单独的 transactionID 是唯一的。如果接收(需要多个请求/响应)交易的第一条消息的服务端接收到不允许其满足上述约束的 transactionID(通常是因为 transactionID 已在使用中),应发送回带有 transactionIdInUse 的 PKIFailureInfo 的 ErrorMessageContent。客户端在交易开始时宜使用 128 位随机数填充 transactionID 字段,以降低服务端使用该 transactionID 的可能性。

- senderNonce 及 recipNonce: 发送方和接收方随机数,用于保护 PKIMessage 免受重放攻击。senderNonce 通常是由发送者生成的 128 位随机数,而 recipNonce 复制交易的前一个消息的 senderNonce。
- messageTime: 包含发送者创建这个消息的时间。可由终端用于校正/检查自己的本地时间以便与中央系统的时间保持一致。
- freeText: 可用于发送人可读的消息给接收者(用任意多种语言)。此序列中的第一种语言表明期望的响应语言。
- generalInfo: 可用于向接收方发送非供人工使用的额外数据。可支持 C.1.2.2~C.1.2.4 定义的 generalInfo 扩展项。

C.1.2.2 隐式确认

终端或 RA 系统利用以下数据结构通知 RA 或 CA 系统,其在收到证书后,将不发送 C.3.15 定义的证书确认消息。

implicitConfirm OBJECT IDENTIFIER ::= {id-it 13}

ImplicitConfirmValue ::= NULL

若 RA 或 CA 系统许可终端或 RA 系统的请求,则应在响应消息的 PKIHeader 头中放置相同的扩展名。若终端或 RA 系统未在响应消息中找到此扩展,则应发送证书确认。id-it 的值见 C.3.16。

C.1.2.3 确认等待时间

CA 或 RA 系统利用以下数据结构通知 RA 系统或终端,在撤销证书和删除交易之前其打算等待证书确认的时间长度。

confirmWaitTime OBJECT IDENTIFIER ::= {id-it 14}

ConfirmWaitTimeValue ::= GeneralizedTime

C.1.2.4 证书模板标识

RA 系统利用以下数据结构通知 CA 系统,其对要申请的证书的要求,如申请签名证书或签名证书和加密证书。

certTemplateID OBJECT IDENTIFIER ::= {1.2.156.10197.6.4.1.1}

CertTemplateIDValue ::= UTF8String

C.1.3 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(C.3.1)
ip	[1]	CertRepMessage, --初始化响应(C.3.2)
cr	[2]	CertReqMessages, --认证请求(C.3.3)
cp	[3]	CertRepMessage, --认证响应(C.3.4)
p10cr	[4]	CertificationRequest,--PKCS # 10 认证请求

kur	[7]	CertReqMessages,	--密钥更新请求(C.3.5)
kup	[8]	CertRepMessage,	--密钥更新响应(C.3.6)
krr	[9]	CertReqMessages,	--密钥恢复请求(C.3.7)
krp	[10]	KeyRecRepContent,	--密钥恢复响应(C.3.8)
rr	[11]	RevReqContent,	--撤销请求(C.3.9)
rp	[12]	RevRepContent,	--撤销响应(C.3.10)
cann	[16]	CertAnnContent,	--证书公告(C.3.11)
rann	[17]	RevAnnContent,	--撤销公告(C.3.12)
crlann	[18]	CRLAnnContent,	--CRL 公告(C.3.13)
pkiconf	[19]	PKIConfirmContent,	--PKI 确认(C.3.14)
nested	[20]	NestedMessageContent,	--嵌套消息(C.1.4)
genm	[21]	GenMsgContent,	--通用消息(C.3.16)
genp	[22]	GenRepContent,	--通用响应(C.3.17)
error	[23]	ErrorMsgContent,	--错误消息(C.3.18)
certConf	[24]	CertConfirmContent,	--证书确认(C.3.15)
pollReq	[25]	PollReqContent,	--轮询请求(C.3.19)
pollRep	[26]	PollRepContent,	--轮询响应(C.3.19)
fr	[27]	RevReqContent,	--证书冻结请求(C.3.20)
fp	[28]	RevRepContent,	--证书冻结响应(C.3.20)
ufr	[29]	RevReqContent,	--证书解冻请求(C.3.21)
ufp	[30]	RevRepContent	--证书解冻响应(C.3.21)
}			

具体类型消息的描述见 C.3。CertificationRequest 的定义见 GM/T 0092。

C.1.4 PKI 消息保护

某些 PKI 消息需要保护完整性。(例如使用已经认证过的公钥证明消息的来源以及消息的完整性。)

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,可采用 PKIX 以外的其他保护方法保护消息,而非采用 PKIProtection。例如,使用 PKCS #7[PKCS7]或 Security Multiparts[RFC 1847]对 PKIMessage 进行封装(当 PKIHeader 信息由外部机制安全地传输时,仅对省略 CHOICE 标签的 PKIBody 进行封装)。许多此类机制要求终端已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。因此,此类方法对于初始注册、密钥恢复等具有“初始”特性的过程不适用。针对此类情况,应使用 PKIProtection。

根据环境不同,PKIProtection 比特串可包含一个消息认证码(MAC)或数字签名,包括下面几种情况。

- 共享密钥信息。在此种情况下,发送方和接收方共享密钥信息。PKIProtection 将包含一个 MAC 值,protectionAlg 如下(见 A.3)。

id-PasswordBasedMac OBJECT IDENTIFIER ::= {1 2 840 113533 7 66 13}

```
PBMPParameter ::= SEQUENCE {
    salt                OCTET STRING,
    owf                 AlgorithmIdentifier,
    --单向函数(OWF)的算法标识
    iterationCount      INTEGER,
    -- OWF 的应用次数
    mac                 AlgorithmIdentifier
    -- MAC 算法标识
}
```

在上述 protectionAlg 中,共享密钥后面追加 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(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 $K \leq H$,密钥则取 BASEKEY 的前 K 比特;如果 $K > H$,则将 BASEKEY 的全部 H 比特作为密钥的前 H 比特,OWF(“1” || BASEKEY)所得结果作为密钥的下一 H 比特,OWF(“2” || BASEKEY)所得结果作为密钥的再下一 H 比特,依此类推,直到得到所有的 K 比特为止(“N”代表数字 N 的 ASCII 字节编码,“||”代表串联)。PBMPParameter 字段在一个交易的所有消息中(例如 ir/ip/certConf/pkiConf)宜保持不变,以降低计算 PasswordBasedMac 的负荷。

- b) 签名。在此种情况下,发送方拥有一个签名密钥对,对 PKI 消息进行签名。PKIProtection 包含签名值,protectionAlg 为一种数字签名算法。
- c) 多重保护。当终端发送一个保护的 PKI 消息到 RA 系统,RA 系统可追加自己的保护(可是 MAC 或签名,取决于 RA 和 CA 系统之间共享的信息和证书)后将消息转发到 CA 系统。这种保护通过将终端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

[使用 PKIMessages,即是一系列 PKIMessage,允许 RA 系统在单个新消息中批处理多个终端的请求。为简单起见,批处理中的所有消息应具有相同的类型(例如,ir)]。如果 RA 系统希望以某种方式修改消息(例如,添加特定的字段值或新的扩展),则 RA 系统可能会自己创建所需的 PKIBody。来自终端的原始 PKIMessage 可包括在 PKIHeader 的 generalInfo 字段中(例如,为了适应 CA 系统希望检查原始终端消息上的 POP 或其他信息的情况)。在这种情况下,要使用的 infoType 为{id-it 15},infoValue 为 PKIMessages(内容应与 PKIBody 中的请求顺序相同)。

C.2 公共数据结构

C.2.1 被申请的证书内容

各种证书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 结构体使得终端或者 RA 系统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,但所有字段都是可选的。

CertTemplate 语法如下:

```
CertTemplate ::= SEQUENCE {
    version          [0]Version          OPTIONAL,
    serialNumber     [1]INTEGER           OPTIONAL,
```

```
signingAlg      [2]AlgorithmIdentifier  OPTIONAL,
issuer          [3]Name                  OPTIONAL,
validity        [4]OptionalValidity     OPTIONAL,
subject         [5]Name                  OPTIONAL,
publicKey       [6]SubjectPublicKeyInfo OPTIONAL,
issuerUID       [7]UniqueIdentifier     OPTIONAL,
subjectUID      [8]UniqueIdentifier     OPTIONAL,
extensions      [9]Extensions           OPTIONAL
}
Version ::= INTEGER
```

C.2.2 加密值

在 PKI 消息中发送加密值时,应符合 GB/T 35276—2017 的 7.4 所定义的数据结构:

```
SM2EnvelopedKey ::= SEQUENCE {
    symAlgID          AlgorithmIdentifier, --对称密码算法标识
    symEncryptedKey   SM2Cipher,          --对称密钥密文
    sm2PublicKey      SM2PublicKey,       --SM2 公钥
    sm2EncryptedPrivateKey BIT STRING    --SM2 私钥密文
}
```

C.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,期望稍后获取结果(对具有该状态的响应,恰当的处理方式可使用
    C.3.19 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
    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),
--不准许或不支持的交易	
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),
--在要求存在时间戳时没有提供(根据策略要求)	
badPOP	(9),
--拥有证明失败	
certRevoked	(10),
--该证书已被撤销	
certConfirmed	(11),
--证书已被确认	
wrongIntegrity	(12),
--完整性无效	
badRecipientNonce	(13),
--无效的收件人随机数	
timeNotAvailable	(14),
--TSA 的时间源不可用	
unacceptedPolicy	(15),
--TSA 不支持请求的 TSA 策略	
unacceptedExtension	(16),
--TSA 不支持请求的扩展名	
addInfoNotAvailable	(17),
--请求的附加信息无法理解或不可用	
badSenderNonce	(18),
--无效的发件人随机数	
badCertTemplate	(19),

```
--无效的证书模板或缺少必填信息
signerNotTrusted          (20),
--未知或不信任的报文签名者
transactionIdInUse        (21),
--交易标识符已在使用中
unsupportedVersion         (22),
--不支持的报文版本
notAuthorized             (23),
--发件人无权提出前述请求或执行前述操作
systemUnavail             (24),
--由于系统不可用,请求无法处理
systemFailure             (25),
--由于系统故障,请求无法处理
duplicateCertReq          (26),
--由于同样的证书已经存在,所以系统不能发放
certFrozen                (27)
--该证书已被冻结
}
PKIStatusInfo ::= SEQUENCE {
    status          PKIStatus,
    statusString    PKIFreeText          OPTIONAL,
    failInfo        PKIFailureInfo      OPTIONAL
}
```

C.2.4 证书标识

CertId 数据结构用于鉴别特定的证书。

CertId 的语法见下：

```
CertID ::= SEQUENCE{
    hash Algorithm          AlgorithmIdentifier,
    issuerNameHash          OCTET STRING,
    --发布者 DN 的杂凑值
    issuerKeyHash           OCTET STRING,
    --发布者公钥的杂凑值
    serialNumber            CertificateSerialNumber
}
```

C.2.5 发布根 CA 公钥

每个根 CA 应能发布其当前公钥,定义了支持这种方法的数据结构。

一般可通过两种方法发布根 CA 系统的公钥:一种是 CA 系统直接发布其自签名证书;另外一种是通过目录服务发布,CA 系统同时发布该自签名证书的杂凑值,以便允许验证 CA 系统证书的完整性。

OOBCert ::= Certificate

该证书中的值域有如下限制:

——证书应是自签名的(即签名可用 SubjectPublicKeyInfo 中的值来验证);

- 主体字段和签发者字段的值应完全相同；
- 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应同时存在，并且具有完全相同的值；
- 所有其他扩展项的值应符合自签名证书的要求（比如，主体和签发者的密钥标识须完全相同）。

```

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

——其目的在于任何安全获取到该杂凑值的用户都能验证该 CA 系统的自签名证书

```

}

```

C.2.6 归档选项

请求者可使用 PKIArchiveOptions 结构体来表明要求 PKI 管理组件归档某私钥。

PKIArchiveOptions 的语法见下：

```

PKIArchiveOptions ::= CHOICE {
    encryptedPrivKey      [0] SM2EnvelopedKey,
    --私钥
    keyGenParameters      [1] KeyGenParameters,
    --允许重新产生私钥的参数
    archiveRemGenPrivKey [2] BOOLEAN
    --如果发送者期望接收者归档其对应请求所生成密钥对中的私钥
    --则设置为真，如果不需要归档，则设置为假
}

```

C.2.7 发布信息

请求者可使用 PKIPublicationInfo 结构体来表明要求 PKI 管理组件发布证书。

PKIPublicationInfo 的语法见下：

```

PKIPublicationInfo ::= SEQUENCE {
    action    INTEGER {
        dontPublish    (0),
        pleasePublish   (1)
    },
    pubInfos SEQUENCE SIZE (1..MAX) OF SinglePubInfo OPTIONAL
}

```

如果 action 项为“dontPublish”（不发布），则 pubInfos 应为空。

C.2.8 拥有证明结构体

为使 RA 和 CA 系统能有效验证终端和密钥对间的绑定是否合法，终端可通过私钥拥有证明 POP 证明自己拥有并能使用与申请证书的公钥相对应的私钥。PKI 在认证交换时可自由选择如何实现 POP（例如定义的方式或其他方式）。

允许终端向 RA 系统提供相关证明，RA 系统收到证明并验证后向 CA 系统说明需要的证明已完

成。某些策略可能不准许此情形(例如,CA 系统在认证过程中可唯一验证 POP)。

终端通过对一个数据签名来证明拥有签名密钥的私钥。

为签名密钥对申请证书,可使用 POPOSigningKey 结构体来证明对签名私钥的拥有。

POPOSigningKey 的语法如下:

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

其中:

——poposkInput 包含要签名的数据,证书模板不包含公钥值和主体名称值时应有此字段;

——algorithmIdentifier 标识用于生成 POP 值的签名算法和相关参数;

——signature 包含生成的 POP 值。如果存在 poposkInput,则根据 poposkInput 的 DER 编码值计算签名;如果没有 poposkInput,则根据 certReq 的 DER 编码值计算签名。

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建立之后使用)
        publicKeyMAC      PKMACValue
        --发送者当前无已鉴别通过的通用名(GeneralName)存在时使用;publicKeyMAC 包含一
        --个基于口令的对公钥的 DER 编码的 MAC 值(使用 PKIHeader 中的 protectionAlg
        --算法)
    },
    publicKey            SubjectPublicKeyInfo
    --取自 CertTemplate
}
```

C.3 特定操作数据结构



C.3.1 初始化请求

初始化请求消息用于终端在 PKI 体系中的最初初始化。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一个或多个证书。利用以下数据结构来进行多个证书的申请时,在具体实现上有多种不同的方式,常见的一种实现方式见 A.5。

```
CertReqMessages ::= SEQUENCE SIZE (1..MAX) OF CertReqMsg
```

```
CertReqMsg ::= SEQUENCE {
    certReq              CertRequest,
    pop                  ProofOfPossession OPTIONAL,
    regInfo              SEQUENCE SIZE(1..MAX) OF AttributeTypeAndValue OPTIONAL
}
```

```
CertRequest ::= SEQUENCE {
    certReqId            INTEGER,
    certTemplate         CertTemplate,
```

```

        controls          Controls OPTIONAL
    }
    Controls ::= SEQUENCE SIZE(1..MAX) OF AttributeTypeAndValue
    AttributeTypeAndValue ::= SEQUENCE {
        type      AttributeType OBJECT IDENTIFIER,
        value     AttributeValue ANY DEFINED BY type
    }
    ProofOfPossession ::= CHOICE {
        raVerified          [0] NULL,
        --当 RA 系统已经验证了请求者拥有相应私钥时使用
        signature           [1] POPOSigningKey,
        --用户产生证书密钥对,并申请签名证书时使用
        keyEncipherment     [2] POPOPrivKey,
        --不使用该域
        keyAgreement        [3] POPOPrivKey
        --不使用该域
    }

```

C.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。利用以下数据结构来进行多个证书的响应时,在具体实现上有多种不同的方式,一种常见的实现方式见 A.5。

```

    CertRepMessage ::= SEQUENCE {
        caPubs          [1] SEQUENCE SIZE (1..MAX) OF Certificate OPTIONAL,
        response         SEQUENCE OF CertResponse
    }
    CertResponse ::= SEQUENCE {
        certReqId        INTEGER,
        --将该响应与相应的请求进行匹配(请求中未指定 certReqId 值时,取值为-1)
        status           PKIStatusInfo,
        certifiedKeyPair  CertifiedKeyPair OPTIONAL,
        rspInfo          OCTET STRING OPTIONAL
        --与[CRMF]中 CertReqMsg 的 regInfo 字段定义 id-regInfo-asciiPairs 相似
    }
    CertifiedKeyPair ::= SEQUENCE {
        certOrEncCert    CertOrEncCert,
        privateKey        [0] SM2EnvelopedKey OPTIONAL,
        publicationInfo   [1] PKIPublicationInfo OPTIONAL
    }
    CertOrEncCert ::= CHOICE {
        certificate       [0] Certificate,
        encryptedCert     [1] SM2EnvelopedKey
    }

```


C.3.3 认证请求

当已存在于 PKI 体系中的组件想获取额外的证书时使用此消息,数据结构同 C.3.1。

C.3.4 认证响应

此消息用于对认证请求进行响应,数据结构同 C.3.2。

C.3.5 密钥更新请求

此消息用于 RA 系统向 CA 系统请求更新已存在的证书(未作废且未过期),数据结构同 C.3.1。该操作也被称作“证书更新”操作,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证书。

C.3.6 密钥更新响应

此消息用于 CA 系统针对密钥更新请求向 RA 系统进行响应,其数据结构同 C.3.2。

C.3.7 密钥恢复请求

此消息用于 RA 系统向 CA 系统请求恢复密钥,数据结构同 C.3.1。



C.3.8 密钥恢复响应

此消息用于 CA 系统针对密钥恢复请求向 RA 系统进行响应。

```
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
}
```

C.3.9 撤销请求

```
RevReqContent ::= SEQUENCE OF RevDetails
RevDetails ::= SEQUENCE {
    certDetails CertTemplate,
    --允许请求者为请求作废的证书指定尽可能多的信息(如:在 serialNumber 未知的情况下)
    revocationReason ReasonFlags OPTIONAL,
    --请求作废的原因
    badSinceDate GeneralizedTime OPTIONAL,
    crlEntryDetails Extensions OPTIONAL
    --请求的 crlEntryExtensions
}
ReasonFlags ::= BIT STRING{
    Unused          (0),
    keyCompromise   (1),
    CACompromise    (2),
    affiliationChanged (3),
    superseded      (4),
```

```

        cessationOfOperation      (5),
        certificateHold            (6)
    }

```

C.3.10 撤销响应

```

RevRepContent ::= SEQUENCE {
    status      SEQUENCE SIZE (1..MAX) OF PKIStatusInfo,
    --与 RevReqContent 中的次序相同
    revCerts [0]SEQUENCE SIZE (1..MAX) OF CertId OPTIONAL,
    --请求作废的证书的 ID(与 status 次序相同)
    crls       [1]SEQUENCE SIZE (1..MAX) OF CertificateList OPTIONAL
    --由此产生的 CRLs(可能不止一个)
}

```

C.3.11 证书公告

此结构体可用于公告证书的存在。

CertAnnContent ::= Certificate

C.3.12 撤销公告

当 CA 系统已撤销或将撤销一个证书时,可利用以下数据结构公告该事件。

```

RevAnnContent ::= SEQUENCE {
    status PKIStatus,
    certId CertId,
    willBeRevokedAt GeneralizedTime,
    badSinceDate GeneralizedTime,
    crlDetails Extensions OPTIONAL
}

```

willBeRevokedAt 字段表示一个新的条目将加入到相应 CRLs 的时间。

C.3.13 CRL 公告

当 CA 系统签发了一个新的 CRL(或一批 CRL)时,可使用以下数据结构宣布该事件。

CRLAnnContent ::= SEQUENCE OF CertificateList

C.3.14 PKI 确认

该消息在协议数据交换中可作为最后的 PKIMessage,此确认消息为可选的。由于 PKIHeader 中已包含所有必需的消息,所以其内容为空。

PKIConfirmContent ::= NULL

C.3.15 证书确认

终端使用此数据结构向 PKI 发送确认接受或拒绝证书的消息,此消息为可选的。

```

CertConfirmContent ::= SEQUENCE OF CertStatus
CertStatus ::= SEQUENCE {
    certHash OCTET STRING,

```

```

certReqId INTEGER,
statusInfo PKIStatusInfo OPTIONAL,
hashAlg [0] AlgorithmIdentifier{DIGEST-ALGORITHM,{...}} OPTIONAL
}

```

可选字段 hashAlg 的使用以便为待确认证书的 signatureAlgorithm 字段中未指定杂凑算法时显式指定杂凑算法,用于计算 certHash 值。

C.3.16 PKI 通用消息

```

InfoTypeAndValue ::= SEQUENCE {
    infoType OBJECT IDENTIFIER,
    infoValue ANY DEFINED BY infoType OPTIONAL
}

```

InfoTypeAndValue 的内容包含但不限于下列值:

```

——{ 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}。

```

GenMsgContent ::= SEQUENCE OF InfoTypeAndValue

```

如果使用 PKI 消息来请求和提供 PKI 相应的信息,那么请求应使用 GenMsg 消息,响应使用 GenRep 消息(见 C.3.17),错误使用 Error 错误消息(见 C.3.18)。上述消息使用基于共享秘密信息的 MAC (如:PasswordBasedMAC)或者其他认证方法(如果终端拥有证书)来进行保护。

在证书撤销请求中使用共享秘密信息的安全机制,见附录 E。关于 InfoTypeAndValue 内容更多的细节及语法描述见附录 F。

C.3.17 PKI 通用响应

```

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

```

C.3.18 错误消息

```

ErrorMsgContent ::= SEQUENCE {
    pKIStatusInfo PKIStatusInfo,
    errorCode INTEGER OPTIONAL,
    errorDetails PKIFreeText OPTIONAL
}

```

C.3.19 轮询请求和响应

以下数据结构用于处理客户端需要轮询服务端以确定未完成交易的状态的情况(即当收到“等待”的 PKI 状态时)。轮询请求和响应消息为可选的消息。

```

PollReqContent ::= SEQUENCE OF SEQUENCE {
    certReqId INTEGER }

```

PollRepContent ::= SEQUENCE OF SEQUENCE {
 certReqId INTEGER,
 checkAfter INTEGER, --以秒计的时间数
 reason PKIFreeText OPTIONAL }

C.3.20 证书冻结

证书冻结请求与响应数据结构同 C.3.9 和 C.3.10。

C.3.21 证书解冻

证书解冻请求与响应数据结构同 C.3.9 和 C.3.10。

附录 D

(资料性)

版本协商

D.1 通则

如果客户端知道服务端支持的协议版本(例如,通过之前的 PKIMessage 交互或某些 PKI 消息之外的手段),则客户端需要使用其和服务端都支持的最高版本发送 PKIMessage;如果客户端不知道服务端支持的版本,则需要使用其支持的最高版本发送 PKIMessage。

如果服务端收到其支持的版本的 PKIMessage,则响应消息的版本应与收到的消息版本相同。如果服务端收到一个其不支持的更高或更低版本的信息,则应发送一个在 unsupportedVersion 字段进行比特设置的错误消息 ErrorMsg(在 pKIStatusInfo 的 failureInfo 字段中)。如果服务端收到的版本高于其支持的最高版本,则错误消息的版本应是服务端支持的最高版本;如果收到的版本低于服务端支持的最低版本,则错误信息中的版本应是服务端支持的最低版本。

如果客户端收到一个带有 unsupportedVersion 字段比特设置的 ErrorMsgContent,并且版本是其所支持的,那么客户端可选择使用该版本重试请求。

客户端和服务端宜采用一定机制来防止降低协议版本的密码攻击。

D.2 与 GB/T 19714—2005 版本服务端对话的客户端

如果客户端在发送本文件版本的消息后,收到带有 GB/T 19714—2005 版本的 ErrorMsgContent,中止当前事务。随后,客户端可使用 GB/T 19714—2005 版本消息重试事务。

如果客户端收到 GB/T 19714—2005 版本的非错误消息,则可使用 GB/T 19714—2005 版本消息继续事务(如果事务尚未完成)。如果客户端不使用 GB/T 19714—2005 版本消息继续事务并且事务未完成,则中止事务并发送一个带有 GB/T 19714—2005 版本的 ErrorMsgContent。

D.3 接收 GB/T 19714—2005 版本消息的服务端

如果服务端收到 GB/T 19714—2005 版本消息,其可能会切换到 GB/T 19714—2005 行为,并用 GB/T 19714—2005 版本消息进行响应。如果服务端不使用 GB/T 19714—2005 版本消息进行响应,那么则发回一个 ErrorMsgContent。

附 录 E
(资料性)
使用“口令短语”

撤销请求宜包含适当的安全机制如认证机制,用于减少拒绝服务攻击成功的概率。对撤销请求进行数字签名能提供所要求的认证机制,但是在某些情况下需要替代机制(例如,私钥已经不能访问,但是终端希望在重新认证另一个密钥对之前请求撤销)。为了满足此情况,如果支持撤销请求并且在撤销之前请求者和响应者之间可建立共享秘密信息,则需要支持对撤销请求的 PasswordBasedMAC(以符合给定环境的本地安全策略)。

在某些环境中使用的一种机制是“Revocation Passphrase”,其中在撤销之前,终端和 CA/RA 系统之间共享一个具有足够熵的值(即,一个相对长的 passphrase 而不是短的 password),此值用于以后验证撤销请求。

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。其在 CMP 消息中的准确使用如下。

- C.3.16 中指定的 OID 以及值可在任何时间通过 GenMsg 消息发送,也可在任何时间任何一个 PKIMessage 的 PKIHeader 的 generalInfo 字段中发送。此消息向相关的 CA/RA 系统传递一个由终端选择的 revocation passphrase(即当使用 SM2EnvelopedKey 时,为 Sm2EncryptedPrivateKey 字段解密后的字节)。
- 如果 CA/RA 系统在 GenMsg 中接收到 revocation passphrase(C.3.16 中指定的 OID 以及值),则构造并发送 GenRep 消息,此消息包含在 C.3.16 中指定的 OID(值为空)。如果 CA/RA 系统在任何一个 PKIMessage 的 PKIHeader 的 generalInfo 字段中接收到 revocation passphrase,则在相应响应 PKIMessage 的 PKIHeader 的 generalInfo 字段中包含该 OID(值为空)。如果 CA/RA 系统由于某种原因不能返回适当的响应消息,则返回状态为“rejection”的错误消息,并可选择给出 failInfo 的原因。
- 撤销请求消息用 PasswordBasedMAC 来保护,用 revocation passphrase 作为密钥。PKIHeader 中的 senderKID 字段可包含密钥标识符用于帮助以后检索正确的 passphrase。

使用以上技术,revocation passphrase 可在任何时候在无需额外消息的情况下进行初始化建立和更新。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)可在 PKIHeader 中包含一个新的 revocation passphrase 以用于验证以后对终端其他证书的撤销请求。在某些情况下,相比在撤销请求消息中暴露 passphrase 的机制,此机制更具优势(暴露 passphrase 可能会引起拒绝服务攻击,即未授权的第三方利用暴露的 passphrase 对终端别的证书的撤销请求进行认证)。由于在请求消息中没有暴露 passphrase,所以不要求在产生撤销请求时总是更新 passphrase(即在不同的时间对不同证书撤销请求的验证可使用同一个 passphrase)。

以上技术可在不使用签名的情况下对撤销请求消息提供强密码保护。通过暴露 revocation passphrase 来对撤销请求进行认证的技术通常不对请求消息字段提供密码保护(因此一个证书的撤销请求可能会被未授权的第三方修改为对该终端另一个证书的撤销请求)。

附录 F

(资料性)

证书管理协议 ASN.1 描述

PKIXCMP {iso(1) 国际标准化组织成员标识(2) 中国(156) 国家密码行业标准化技术委员会(10197) 标准体系(6) 基础设施(4) 1(《网络安全技术 公钥基础设施 证书管理协议》)}

--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), cmp2024(103) },

sender GeneralName,

--标识发送者

recipient GeneralName,

--标识预期的接收者

messageTime [0] GeneralizedTime OPTIONAL,

--产生此消息的时间

protectionAlg [1] AlgorithmIdentifier OPTIONAL,

--用于计算保护比特的算法

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[RFC 3629]编码的文本

PKIBody ::= CHOICE { --特定的消息体元素

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
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
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
fr	[27]	RevReqContent,	--证书冻结请求
fp	[28]	RevRepContent,	--证书冻结响应
ufr	[29]	RevReqContent,	--证书解冻请求
ufp	[30]	RevRepContent,	--证书解冻响应

}
 PKIProtection ::= BIT STRING
 ProtectedPart ::= SEQUENCE {
 header PKIHeader,
 body PKIBody
 }
 id-PasswordBasedMac OBJECT IDENTIFIER ::= {1 2 840 113533 7 66 13}
 PBMPParameter ::= 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
 --这样可减少拒绝服务攻击
 owf AlgorithmIdentifier,
 --单向函数算法 ID(例如,SM3)
 iterationCount INTEGER,
 --OWF 的应用次数
 --为了减少拒绝服务攻击的可能性,实现可能会限制


```

--此整数值为可接受的大小以适配环境中
mac                      AlgorithmIdentifier
--MAC 算法 ID (例如, SM4_MAC 或者 SM3_HMAC)
}

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),
        --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),
        --不准许或不支持的交易
    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),
        --在要求存在时间戳时未提供 (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
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
certFrozen           (27)
    --该证书已被冻结
}
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
}

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 ,则此项填-1）
    status        PKIStatusInfo,
    certifiedKeyPair CertifiedKeyPair      OPTIONAL,
    rspInfo       OCTET STRING             OPTIONAL
    --类似于 RFC 4211 中 CertReqMsg 结构中为 regInfo 定义的 id-regInfo-utf8Pairs 字符串
}

CertifiedKeyPair ::= SEQUENCE {
    certOrEncCert CertOrEncCert,
    privateKey     [0] SM2EnvelopedKey    OPTIONAL,
    publicationInfo [1] PKIPublicationInfo  OPTIONAL
}

CertOrEncCert ::= CHOICE {
    certificate      [0] CMPCertificate,
    encryptedCert    [1] SM2EnvelopedKey
}

KeyRecRepContent ::= SEQUENCE {
    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)
}

```

```

CertAnnContent ::= CMPCertificate

```

```

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)
}

```

```

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
    hashAlg [0] AlgorithmIdentifier OPTIONAL
    --用于计算 certHash 的杂凑算法
    --要确认的证书的 AlgorithmIdentifier 字段指定了杂凑算法的情况下,不使用此字段
}

```

```

PKIConfirmContent ::= NULL

```

```

InfoTypeAndValue ::= SEQUENCE {
    infoType      OBJECT IDENTIFIER,
    infoValue     ANY DEFINED BY infoType OPTIONAL
}

```

--实例 InfoTypeAndValue 的内容包含但不限于下列值

--(去掉此 ASN.1 模型的注释,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}
--   KeyPairParamRepValue ::= AlgorithmIdentifier
-- id-it-revPassphrase OBJECT IDENTIFIER ::= {id-it 12}
--   RevPassphraseValue ::= SM2EnvelopedKey
-- 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-supplLangTags OBJECT IDENTIFIER ::= {id-it 16}
--   SupplLangTagsValue ::= SEQUENCE OF UTF8String
-- certTemplateID OBJECT IDENTIFIER ::= {1.2.156.10197.6.4.1.1}
--   CertTemplateIDValue ::= UTF8String

```

--其中

```

-- id-pkix OBJECT IDENTIFIER ::= {iso(1) identified-organization(3)
--   dod(6) internet(1) security(5) mechanisms(5) pkix(7)}

```

--并且

```

-- id-it OBJECT IDENTIFIER ::= {id-pkix 4}

```

--此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

--终端、RA 系统或者 CA 系统均可发送该消息(取决于消息的内容)。通常忽略 GenMsg 中 InfoTypeAndValue 的可选参数 infoValue(即其仅应用于相关的 GenRep 消息中)。接收方可忽略不能识别的对象标识符

GenRepContent ::= SEQUENCE OF InfoTypeAndValue

--接收方可忽略其不能识别的 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,
 checkAfter INTEGER, --以秒为单位的时间
 reason PKIFreeText OPTIONAL

}

参 考 文 献

[1] GB/T 19771 网络安全技术 公钥基础设施 PKI 组件最小互操作规范

[2] GB/T 25056—2018 信息安全技术 证书认证系统密码及其相关安全技术规范

[3] GM/T 0092 基于 SM2 算法的证书申请语法规范

[4] PKCS7 PKCS #7: Cryptographic Message Syntax

[5] PKCS10 PKCS #10: Certification Request Syntax Specification

[6] RFC 1847 Security Multiparts for MIME: Multipart/Signed and Multipart/Encrypted

[7] RFC 2510 Internet X.509 Public Key Infrastructure Certificate Management Protocols

[8] RFC 3629 UTF-8, a transformation format of ISO 10646

[9] RFC 4211 Internet X.509 Public Key Infrastructure Certificate Request Message Format (CRMF)

[10] RFC 4511 Lightweight Directory Access Protocol (LDAP): The Protocol

[11] RFC 4512 Lightweight Directory Access Protocol (LDAP): Directory Information Models

[12] RFC 4517 Lightweight Directory Access Protocol (LDAP): Syntaxes and Matching Rules

[13] RFC 4520 Internet Assigned Numbers Authority (IANA) Considerations for the Lightweight Directory Access Protocol (LDAP)

[14] RFC 5652 Cryptographic Message Syntax (CMS)

[15] RFC 6082 Deprecating Unicode Language Tag Characters: RFC 2482 is Historic

[16] RFC 9480 Certificate Management Protocol (CMP) Updates