



中华人民共和国国家标准

GB/T 19713—2025

代替 GB/T 19713—2005

网络安全技术 公钥基础设施 在线证书状态协议

Cybersecurity technology—Public key infrastructure—
Online certificate status protocol

2025-02-28 发布

2025-09-01 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 1

5 总则 2

 5.1 概述 2

 5.2 请求 2

 5.3 响应 2

 5.4 异常情况 3

 5.5 时间语义 4

 5.6 预产生响应 4

 5.7 OCSP 签名机构的委托 4

 5.8 CA 密钥泄漏 4

6 功能要求 4

 6.1 证书内容要求 4

 6.2 签名响应的接收要求 4

7 具体语法 5

 7.1 约定 5

 7.2 请求 5

 7.3 响应 7

 7.4 扩展 11

附录 A（规范性） OCSP 请求和响应的 ASN.1 语法规范 15

附录 B（规范性） 基于 HTTP 的 OCSP 请求和响应 24

附录 C（资料性） OCSP 请求和响应 ASN.1 语法消息示例 26

附录 D（资料性） 安全考虑 34

参考文献 36

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 19713—2005《信息技术 安全技术 公钥基础设施 在线证书状态协议》，与 GB/T 19713—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“本标准适用于各类基于公开密钥基础设施的应用程序和计算环境”为“本文件适用于公钥基础设施的建设以及基于在线证书状态协议的安全应用等”(见第1章,2005年版的第1章)；
- b) 在“通则”中增加了 OCSP 协议中各方之间的关系图(见 5.1,2005 年版的 5.1)；
- c) 更改了“响应的哈希签名”为“响应的数字签名”[见 5.3 b),2005 年版的 5.3 f)]；
- d) 更改了 revoked(已撤销)状态的使用范围,允许对从未签发过的证书使用此响应状态[见 5.3 d),2005 年版的 5.3]；
- e) 增加了对未签发证书状态请求的响应要求[见 5.3 e)]；
- f) 更改了 unauthorized(未授权)错误响应的使用范围(见 5.4,2005 年版的 5.4)；
- g) 增加了 revocationTime(撤销时间)语义的定义(见 5.5)；
- h) 增加了 SM2、SM3 算法的支持(见 7.1 和 7.2)；
- i) 增加了 OCSP ASN.1 语法中 Signature、Extensions、CertificateSerialNumber、SubjectPublicKeyInfo、Name、AlgorithmIdentifier 和 CRLReason 结构的定义(见 7.1)；
- j) 增加了轻量级 OCSP 请求语法的注解(见 7.2.2)；
- k) 增加了轻量级 OCSP 协议对时间的要求(见 7.3.2.1)；
- l) 更改了“本地配置的 OCSP 签名权威实体中包含了与待验证状态的证书相匹配的证书”为“本地配置的 OCSP 响应者证书与 OCSP 响应者证书相匹配”(见 7.3.2.2.2,2005 年版的 7.3.2.2)；
- m) 增加了轻量级 OCSP 环境下授权响应者的撤销状态检查方法[见 7.3.2.2.3 d)]；
- n) 增加了“7.3.2.3 基础响应”,并阐明了 ResponderID 字段对应于 OCSP 响应者签名证书(见 7.3.2.3)；
- o) 增加了轻量级 OCSP 响应中对 OCSPResponse 结构的要求[见 7.3.2.3 e)]；
- p) 增加了“7.3.2.2.4 证书状态发布”,对 OCSP 响应者获取证书状态应遵循的标准进行了描述(见 7.3.2.2.4)；
- q) 删除了强制的密码算法和可选的密码算法(见 2005 年版的 7.4)；
- r) 更改了 Nonce 的 ASN.1 语法,并规定了 Nonce 的长度范围(见 7.4.2,2005 年版的 7.5.1)；
- s) 更改了 CRL 条目扩展应遵循的标准(见 7.4.6,2005 年版的 7.5.5)；
- t) 增加了“优先使用的签名算法”扩展,该扩展可包含在请求消息中,以指定请求者希望响应者使用的签名算法,建议优先算法使用 SM3WithSM2(见 7.4.8)；
- u) 增加了“扩展撤销定义”扩展,该扩展表明响应者支持对 5.3 中定义的未签发证书的“revoked(已撤销)”响应的扩展使用(见 7.4.9)；
- v) 更改了使用 ASN.1 的 2008 语法的 ASN.1 模块,增加支持使用 SM2、SM3 算法(见附录 A,2005 年版的附录 B)；增加了轻量级 OCSP ASN.1 的语法规则,并增加支持使用 SM2、SM3 算法(见附录 A)；
- w) 增加了轻量级 OCSP 请求及响应构造(见附录 B.2)；
- x) 更正正文的“安全考虑”为附录 D,并补充完善了内容(见附录 D,2005 年版的第 8 章)。

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:普华诚信信息技术有限公司、上海信息安全基础设施研究中心有限责任公司、上海市数字证书认证中心有限公司、北京数字认证股份有限公司、郑州信大捷安信息技术股份有限公司、深圳市电子商务安全证书管理有限公司、中电科网络安全科技股份有限公司、河南金盾信安检测评估中心有限公司、国家密码管理局商用密码检测中心、格尔软件股份有限公司、三六零数字安全科技集团有限公司、数安时代科技股份有限公司、华为技术有限公司。

本文件主要起草人:梁佐泉、顾青、田文晋、王亚红、冯四风、高五星、张子鸣、付丽丽、王志威、黄成杭、赵艳红、石韶博、陈萃祺、赵鹰侠、张永强、刘为华、郑会涛、岳小阳、梁宏、张绍博、郑强、张志磊、杜志强、曾光。

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 19713—2005;

——本次为第一次修订。

网络安全技术 公钥基础设施 在线证书状态协议

1 范围

本文件给出了一种无需请求证书撤销列表(CRL)即能查询数字证书状态的机制,即在线证书状态协议,包括在线证书状态协议的协议内容、语法规范。

本文件适用于公钥基础设施的建设以及基于在线证书状态协议的安全应用等。

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 16263.1 信息技术 ASN.1 编码规则 第 1 部分:基本编码规则(BER)、正则编码规则(CER)和非典型编码规则(DER)规范

GB/T 19714—2005 信息技术 安全技术 公钥基础设施 证书管理协议

GB/T 20518—2018 信息安全技术 公钥基础设施 数字证书格式

GB/T 25069 信息安全技术 术语

GB/T 32915 信息安全技术 二元序列随机性检测方法

GB/T 33560—2017 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 界定的以及下列术语和定义适用于本文件。

3.1

请求者 requester

申请在线证书状态查询服务的实体或设备。

3.2

响应者 responder

提供在线证书状态查询服务的实体或设备。

3.3

在线证书状态协议 online certificate status protocol; OCSP

一种无需请求证书撤销列表(CRL)即能查询数字证书状态的协议。

4 缩略语

下列缩略语适用于本文件。

CA: 证书认证机构(Certification Authority)

- HTTP:超文本传输协议(Hyper Text Transfer Protocol)
- LDAP:轻量级目录访问协议(Lightweight Directory Access Protocol)
- OID:对象标识符(Object ID)
- PKI:公钥基础设施(Public Key Infrastructure)
- SMTP:简单邮件传输协议(Simple Mail Transfer Protocol)
- URI:统一资源描述符(Uniform Resource Identify)
- URL:统一资源定位符(Uniform Resource Locator)

5 总则

5.1 概述

OCSP 能为需要及时获得证书状态信息的应用程序提供撤销状态;与查询 CRL 方式相比,OCSP 能提供更实时的撤销状态信息,以及附加的状态信息,可作为查询 CRL 的替代或补充方法。

本文件在常规 OCSP 基础上,增加了轻量级 OCSP,以适应超大规模或者需轻量级解决方案以最大限度减少带宽和请求者/响应者处理能力的 PKI 环境下的应用需求。

OCSP 请求者、OCSP 响应者以及 CA 三者之间的关系见图 1,OCSP 请求者向 OCSP 响应者发送证书验证请求,OCSP 响应者接受请求后返回响应数据,CA 向 OCSP 响应者发布证书状态。本文件对 OCSP 请求者和 OCSP 响应者之间需要交换的数据和数据格式进行了描述。



图 1 OCSP 协议图

5.2 请求



OCSP 请求包含的数据内容及响应者对请求数据的具体要求如下。

- a) OCSP 请求包含以下数据：
 - 1) 协议版本；
 - 2) 服务请求；
 - 3) 待验证证书标识符；
 - 4) OCSP 响应者可选扩展信息,比如:OCSP 请求者的签名、随机数。
- b) 在接收到请求时,OCSP 响应者应确定：
 - 1) 报文格式是否正确；
 - 2) 响应者是否配置了所要求的服务；
 - 3) 请求是否包含了响应者需要的信息。

如果上述任一条件不满足,OCSP 响应者将返回一个错误信息;反之,将返回一个明确的响应。

5.3 响应

确定的 OCSP 响应包含的数据内容及对响应数据的具体要求如下。

- a) 所有确定的响应报文都应进行数字签名,用于签名的密钥应属于下列一种：
 - 1) 签发待验证证书的 CA 密钥；
 - 2) CA 指定的响应者(即授权的响应者,见 7.3.2.2)的密钥,该响应者拥有一个 CA 直接签发的带有扩展密钥用法 id-kp-OCSPSigning 的证书,表明响应者可为该 CA 签发 OCSP 响

应,其中 id-kp-OCSPSigning 应符合 GB/T 20518—2018 中 5.2.4.2.5 的规定;

3) 可信赖的响应者密钥,即请求者信任该响应者的密钥。

b) 响应报文由如下内容组成:

- 1) 响应语法的版本;
- 2) 响应者的标识符;
- 3) 生成响应的的时间;
- 4) 对请求中每个证书的响应[详见 c)];
- 5) 可选择的扩展;
- 6) 签名算法的 OID;
- 7) 响应的数字签名。

c) 对请求中每个证书的响应由如下内容组成:

- 1) 待验证证书标识符;
- 2) 证书状态值;
- 3) 响应有效间隔;
- 4) 可选的扩展。

d) 本文件对证书状态值定义了以下响应标识符。

- 1) 在用(good):表示证书是有效的在用证书。此响应表示在其有效期内所请求证书序列号的证书没有被撤销,但并不一定意味着该证书曾经被签发过,或产生响应的的时间是在证书有效性期内。另外,响应扩展可提供关于证书状态信息的附加声明,例如已签发、有效期等。
- 2) 已撤销(revoked):表示证书已被冻结(撤销原因是冻结)或永久的撤销。如果相关联的 CA 没有签发所请求证书的记录,也可返回此状态。
- 3) 未知(unknown):表示响应者不能鉴别待验证状态的证书。通常是因为该响应者无法识别验证请求所包含的颁发者。

注:“已撤销”状态表示拒绝所请求证书序列号的证书,而“未知”状态表示响应者无法确定状态,从而允许请求者决定是否要尝试其他状态信息源(例如 CRL)。“已撤销”响应适用于未签发的证书,其中响应者的目的是引导请求者拒绝证书,而不是尝试其他状态信息源。例如,响应者可能不知道请求序列号是否已签发的证书,或者在响应者提供预先产生的响应时,不能为所有未签发的证书序列号提供签名响应。

e) 当响应者向未签发证书的状态请求发送“已撤销”响应时,响应者应在响应中包含扩展撤销定义(见 7.4.9),从而表明 OCSP 响应者支持“已撤销”状态的扩展定义,以涵盖未签发的证书。另外,对于未签发证书,应在 SingleResponse 结构字段(见 7.3.1)中明确以下内容:

- 1) 应明确指出撤销原因是冻结;
- 2) 应明确指出撤销时间是 1970 年 1 月 1 日;
- 3) 不应包括 CRL 引用扩展(见 7.4.3)或任何 CRL 条目扩展(见 7.4.6)。

见 RFC 6960 中的 2.2。

5.4 异常情况

如果出现异常,OCSP 响应者可返回错误消息,并且不需要对错误消息进行签名。错误分为以下几种情况。

- a) 请求格式错误(malformedRequest):OCSP 响应者接收到的请求不符合 OCSP 语法。
- b) 内部错误(internalError):OCSP 响应者处于非协调的工作状态,应向另一个响应者再次进行查询。
- c) 稍后重试(tryLater):OCSP 响应者正处于运行状态,不应返回所请求证书的状态,即表明存在所需的服务,但是暂时不能响应。

- d) 应对请求签名(sigRequired):响应者要求请求者对请求签名。
- e) 请求未被授权(unauthorized):该查询是由未授权请求者向响应者提出的,或者响应者没有能力进行授权响应。

5.5 时间语义

本文件中定义的响应可包含 thisUpdate、nextUpdate、producedAt 和 revocationTime 四个时间,这四个时间的语义分别如下。

- a) thisUpdate:此次更新时间,响应者确认证书状态的更新时间。
- b) nextUpdate:下次更新时间,表示证书在此时间之前,状态是正常的,并且在此时间可再次获得证书状态更新的信息。
- c) producedAt:签发时间,OCSP 响应者产生该响应的时间。
- d) revocationTime:撤销时间,证书被撤销或者冻结的时间。见 RFC 6960 中的 2.4。

5.6 预产生响应

为说明某一特定时间内证书的状态,OCSP 响应者可预先生成签名响应。签名响应中的 thisUpdate 字段表示证书状态被认为是合法的时间;nextUpdate 字段表示证书状态再次更新的时间;produceAt 字段表示产生响应的时间。

5.7 OCSP 签名机构的委托

证书颁发者通过签发包含扩展密钥用法为 id-kp-OCSPSigning 的 OCSP 签名者证书来指派 OCSP 签名机构,此证书应由认可的 CA 直接签发给响应者,见 7.3.2.2.1。

5.8 CA 密钥泄漏

如果 OCSP 响应者知道某个特定 CA 的私钥已被泄漏,则它可为该 CA 发布的所有证书返回已撤销状态。

6 功能要求

6.1 证书内容要求

为向 OCSP 请求者提供 OCSP 响应者的访问位置,CA 应在证书扩展项中提供用于访问 OCSP 响应者的授权机构访问(AIA),或者在 OCSP 请求者本地配置 OCSP 响应者的访问地址,其中证书扩展应符合 GB/T 20518—2018 中 5.2.4 的相关要求。

支持 OCSP 服务(不管是本地配置还是由授权的 OCSP 响应者提供)的 CA,CA 都应在证书的授权机构访问的证书扩展项中包括一个 OID 值为 id-ad-ocsp 的访问方法和访问 OCSP 响应者的 URI。

请求证书中的访问地址值定义了访问 OCSP 响应者的信息传输方式(如 HTTP),该值中可能包含其他信息(如一个 URL)。

6.2 签名响应的接收要求

在将证书的签名响应视为有效之前,OCSP 请求者应确认:

- a) 响应中所鉴别的证书与对应请求中所查验的证书一致;
- b) 响应方的签名是有效的;
- c) 响应方的签名者身份与请求的预定接收者一致;
- d) 签名者已被授权为查验证书提供签名响应;

- e) 指明证书状态的时间(thisUpdate)为当前最近的时间;
- f) 如果设置了 nextUpdate 字段,此时间晚于请求者当前时间。

7 具体语法

7.1 约定

本文件采用抽象语记法 ASN.1 对 OCSP 请求、响应的信息项进行编码,组成特定的请求、响应数据结构,ASN.1 应符合 GB/T 16263.1 的定义。完整的 OCSP 协议语法规则应符合附录 A 的规定,基于 HTTP 的 OCSP 请求格式和响应格式应符合附录 B 的规定,相关示例见附录 C。如果无特殊说明,默认使用 ASN.1 显式标记。

OCSP ASN.1 语法中的 signature、Extensions、CertificateSerialNumber、SubjectPublicKeyInfo、Name、AlgorithmIdentifier 和 CRLReason 等结构应符合 GB/T 20518—2018 中 5.2 和 5.3.4 的定义。

本文件中的 SM2 算法标识应符合 GB/T 33560—2017 中附录 A 的要求。signature 域签名的结果应按照 ASN.1 编码成 BIT STRING 类型并保存在签名值域内。如果签名算法为 SM2, SM2 密码算法签名数据格式应符合 GB/T 35276—2017 中 7.3 的要求。

7.2 请求

7.2.1 OCSP 请求语法

请求的 ASN.1 语法应遵循如下规定。根据所使用的传输机制(HTTP、SMTP、LDAP 等),实际的消息格式可能会发生相应的变化。

OCSP 请求的 ASN.1 数据结构(完整的 OCSPRequest 结构见附录 A 中的 A.1)为:

$$\text{OCSPRequest} ::= \text{SEQUENCE}\{\text{tbsRequest} \quad \text{TBSRequest}, \text{optionalSignature} \quad [0] \quad \text{EXPLICIT Signature OPTIONAL}\}$$

OCSPPRequest 结构中各域的含义是：

- tbsRequest 域包括查验证书和请求者等信息,是 OCSP 请求中签名值所匹配的原文信息(该签名值为可选项);
- optionalSignature 域包含签名算法中的算法标识符和任何相关的算法参数、签名中的签名值、响应者验证签名所需的客户端证书(证书为可选项,通常不包括客户端的根证书)。

TBSRequest	::=	SEQUENCE{	
version		[0]	EXPLICIT Version DEFAULTv1,
requestorName		[1]	EXPLICIT GeneralName OPTIONAL,
requestList			SEQUENCE OF Request,
requestExtensions		[2]	EXPLICIT Extensions OPTIONAL}

TBSRequest 结构中各域的含义是:

- version 域表示协议版本,依据本文件的请求消息协议版本为 v1(0);
- requestorName 域表示 OCSP 请求者的名称,为可选项;
- requestList 域包含一个或多个证书状态请求;
- requestExtensions 域是可选项,包括适用于 reqCert 域中的请求扩展项(见 7.4)。

```
Signature ::= SEQUENCE{
    signatureAlgorithm      AlgorithmIdentifier,
    signature               BIT STRING,
    certs                   [0] EXPLICIT SEQUENCE OF Certificate OPTIONAL}
```

```
Version ::= INTEGER{v1(0)}
```

```
Request ::= SEQUENCE{
    reqCert                 CertID,
    singleRequestExtensions [0] EXPLICIT Extensions OPTIONAL}
```

Request 结构中各域的含义是：

- reqCert 域包含目标证书的标识符；
- singleRequestExtensions 域是可选项，包括适用于单个证书状态请求的扩展项（见 7.4）。

```
CertID ::= SEQUENCE{
    hashAlgorithm           AlgorithmIdentifier
                           {DIGEST-ALGORITHM, {...}},
    issuerNameHash          OCTET STRING,
    issuerKeyHash           OCTET STRING,
    serialNumber            CertificateSerialNumber
}
```

CertID 结构中各域的含义是：

- hashAlgorithm 域是用来生成 issuerNameHash 值和 issuerKeyHash 值的杂凑算法；
- issuerNameHash 域是颁发者证书主题(DN)的杂凑值，该值应根据所查验证书中的颁发者名称字段的 DER 编码进行杂凑计算；
- issuerKeyHash 域是颁发者证书公钥的杂凑值，该值应根据颁发者证书中主体公钥字段的值（不包括标签和长度）进行杂凑计算；
- serialNumber 域是请求其状态的证书的序列号。

7.2.2 OCSP 请求语法的注解

请求语法的注解如下。

- a) 在 CertID 结构中，本文件宜同时使用 CA 名称的杂凑值和 CA 公钥的杂凑值对颁发者进行标识，其主要原因是不同的 CA 可能使用相同的名称（虽然推荐但不强制名称的唯一性），但两个 CA 的公开密钥是不可能相同的，除非二者共享私钥，或者一方的密钥发生泄漏。
- b) requestExtensions 扩展域为可选项，不应将其设置为关键标志，有关扩展域的介绍见 7.4。对于不能识别的扩展域宜忽略（除非它们有关键性的标志并且不被理解）。
- c) 请求者可选择对 OCSP 请求进行签名，签名时以 tbsRequest 结构作为原文进行签名。如果请求已签名，请求者应在 requestorName 域中指定其名称；对于已签名的请求，请求者可在签名的证书字段中包含 OCSP 响应者验证请求者签名所需要的证书，signature 域的要求见 7.1。
- d) 在轻量级 OCSP 时，请求者可简化请求（轻量级 OCSP ASN.1 的语法规则依据附录 A 中的

A.2),简化的结构如下:

- 1) OCSPRequest 结构在 OCSPRequest.RequestList 域中只包含一个请求;
- 2) OCSPRequest 结构中不宜包含 singleRequestExtensions 域,如果包含该结构,宜仅包含一次性随机数(Nonce)扩展(id-pkix-ocsp-nonce);
- 3) 请求者优先使用 SM3 作为 CertID.issuerNameHash 值和 CertID.issuerKeyHash 值的杂凑算法;
- 4) 请求者不宜发送签名的 ocsdp 请求,因为响应者可能会忽略 OCSP 请求上的签名。当请求不签名时,请求者不应在 OCSPRequest 结构中包含 requestorName 域,但 OCSP 响应者可支持即包含 requestorName 域的未签名 OCSP 请求;
- 5) 如果请求者发送签名的 OCSP 请求,请求者应在 OCSPRequest.requestorName 域中指定其名称。

见 RFC 5019 中的 2.2.1。

7.3 响应

7.3.1 OCSP 响应

响应的 ASN.1 语法应遵循如下规定。根据所使用的传输机制(HTTP、SMTP、LDAP 等),实际的消息格式可能会发生相应的变化。

OCSP 响应的 ASN.1 结构(完整的 OCSPResponse 结构见附录 A 中的 A.1)为:

```
OCSPResponse ::= SEQUENCE{
    responseStatus          OCSPResponseStatus,
    responseBytes            [0] EXPLICIT ResponseBytes OPTIONAL}

OCSPResponseStatus ::= ENUMERATED{
    successful              (0), -- 响应被有效确认
    malformedRequest        (1), -- 请求格式错误
    internalError           (2), -- 内部错误
    tryLater                (3), -- 稍候重试
                           -- (4)未使用
    sigRequired             (5), -- 应对请求签名
    unauthorized            (6)  -- 请求未被授权
}
```



OCSPResponse 结构的各域含义是:

- OCSP 响应中至少应包含一个 responseStatus 域,如果 responseStatus 域的值出现了上述任一异常情况时,响应中则无需设置 responseBytes 域;
- responseBytes 结构由一个响应类型的对象标识符和具体响应报文数据组成,该具体响应报文编码为 OCTET STRING。

```
ResponseBytes ::= SEQUENCE{
    responseType          OBJECT IDENTIFIER,
    response               OCTET STRING}
对于基本 OCSP 响应,responseType 为 id-pkix-ocsp-basic。
```

OCSP 响应者应能产生 id-pkix-ocsp-basic 类型的响应,相应地,OCSP 请求者也应能接收并处理此类响应。

id-pkix-ocsp OBJECT IDENTIFIER ::= {id-ad-ocsp}
id-pkix-ocsp-basic OBJECT IDENTIFIER ::= {id-pkix-ocsp 1}
response 字段的值应为 BasicOCSPResponse 结构的 DER 编码。

BasicOCSPResponse ::= SEQUENCE{
 tbsResponseData ResponseData,
 signatureAlgorithm AlgorithmIdentifier,
 signature BIT STRING,
 certs [0] EXPLICIT SEQUENCE OF Certificate OPTIONAL}

BasicOCSPResponse 结构中各域的含义是:

- ResponseData 域为数字签名的输入,采用 ASN.1 DER 编码;
- signature 域保存对 ResponseData 结构进行签名的结果,具体要求见 7.1;
- signatureAlgorithm 域为签名使用的算法,具体要求见 7.1;
- Certs 域为响应者证书,方便 OCSP 请求者验证响应者签名,如果不包含证书,则不包括证书域。

ResponseData ::= SEQUENCE{
 version [0] EXPLICIT Version DEFAULT v1,
 responderID ResponderID,
 producedAt GeneralizedTime,
 responses SEQUENCE OF SingleResponse,
 responseExtensions [1] EXPLICIT Extensions OPTIONAL}

ResponderID ::= CHOICE{
 byName [1] Name,
 byKey [2] KeyHash}
KeyHash ::= OCTET STRING

ResponderID 结构中的 KeyHash 为响应者公开密钥的杂凑值(不包括标签和长度),与 BasicOCSPResponse 中 signatureAlgorithm 所使用的杂凑算法保持一致。

SingleResponse ::= SEQUENCE{
 certID CertID,
 certStatus CertStatus,
 thisUpdate GeneralizedTime,
 nextUpdate [0] EXPLICIT GeneralizedTime OPTIONAL,
 singleExtensions [1] EXPLICIT Extensions OPTIONAL}

CertStatus ::= CHOICE{
 good [0] IMPLICIT NULL,

revoked [1] IMPLICIT RevokedInfo,
unknown [2] IMPLICIT UnknownInfo}

RevokedInfo ::= SEQUENCE {
 revocationTime GeneralizedTime,
 revocationReason [0] EXPLICIT CRLReason OPTIONAL}

UnknownInfo ::= NULL

7.3.2 OCSP 响应的注解

7.3.2.1 时间

响应结构可包含 thisUpdate、nextUpdate、producedAt 和 revocationTime 四个时间字段域,这四个字段的定义见 5.5。GeneralizedTime 字段的格式应符合 GB/T 20518—2018 中 5.2.3.5.4 的要求。OCSP 响应中时间字段应遵循如下要求。

- a) thisUpdate 和 nextUpdate 两个字段定义了有效时间间隔,该时间间隔和 CRL 中的 {thisUpdate,nextUpdate} 间隔相对应。响应中若 nextUpdate 值比本地系统时间早,则响应无效;响应中 thisUpdate 值比本地系统时间晚,则该响应也无效。
- b) 如果未设置 nextUpdate 字段,则响应者认为比较新的响应始终可用。
- c) 在轻量级 OCSP 环境下,请求者的请求信息不包含一次性随机数(Nonce),且请求者忽略响应中的 Nonce 为了确保 OCSP 响应是最新的,请求者对响应时间进行以下处理:
 - 1) 为了确保 OCSP 响应是最新的,请求者检查 nextUpdate 字段是否存在,并且确保当前时间(以 GMT 时间表示)介于 thisUpdate 和 nextUpdate 时间之间,如果 nextUpdate 字段不存在,则请求者拒绝响应;
 - 2) 请求者允许在 nextUpdate 之后配置一个小的容忍期来接受响应,以处理相对于响应者和高速缓存的微小时钟差异,此容忍期根据调用应用程序环境的时间同步技术的准确性和精度来选择。

见 RFC 5019 中的第 4 章。

7.3.2.2 授权的响应者

7.3.2.2.1 授权响应者的证书要求

签署证书状态信息的密钥不必与签发证书的密钥相同,但签署状态信息的实体应获得授权,因此,证书的颁发者应执行以下操作之一:

- a) 证书的颁发者直接对 OCSP 响应签名;
- b) 授权另一个实体对 OCSP 响应签名。

授权响应者的证书应由请求中标识的 CA 直接签发,且签发证书的 extendedKeyUsage 扩展中应包含 id-kp-OCSPSigning 扩展密钥用法,id-kp-OCSPSigning 的定义为:

id-kp-OCSPSigning OBJECT IDENTIFIER ::= {id-kp 9}

CA 签发 OCSP 响应者证书的密钥,应与待验证证书的颁发者证书密钥相同,因此,依赖于 OCSP 响应的系统应确认 OCSP 响应者证书的颁发者与待验证证书的颁发者一致,应验证其颁发者密钥相同。

7.3.2.2.2 请求者对授权响应者证书的检查

请求者通过在本地配置一个或多个 OCSP 签名权威实体以及信任这些权威实体的 CA,可检测并使用 id-kp-OCSPSigning 值(见 7.3.2.2.1),实现对授权响应者证书的验证。授权响应者证书应符合以下任意条件,如果不符合,请求应被拒绝:

- a) 本地配置的 OCSP 响应者证书与 OCSP 响应者证书相匹配;
- b) 是签发待验证证书的 CA 证书;
- c) 在 extendedKeyUsage 扩展中含有 id-ad-ocspSigning 值,并且由签发待验证证书的 CA 签发。

7.3.2.2.3 授权响应器的撤销检查

OCSP 响应器可为一个或多个 CA 提供状态信息,因此,OCSP 请求者需要检查授权响应器的证书状态。CA 可选择以下任一方法处理此问题。

- a) CA 可指定 OCSP 请求者在响应器证书的整个生存期内信任该响应器。CA 通过在响应器证书中包含 id-pkix-ocsp-nocheck 扩展完成指定,该扩展属于非关键性的扩展,值可为空。另外,对于证书的有效期,上述响应器密钥的泄密同签发 CRL 的 CA 密钥的泄密所带来的后果一样严重,因此,CA 可选择签发一种有效期很短并且经常更新的证书,即短生命周期的证书,此项定义为:

id-pkix-ocsp OBJECT IDENTIFIER ::= { id-ad-ocsp }

id-pkix-ocsp-nocheck OBJECT IDENTIFIER ::= { id-pkix-ocsp 5 }
- b) CA 可指定响应器证书撤销状态的检查方法。如果指定 CRL 检查方式,则需使用 CRL 分发点;如果指定其他检查方式,则需使用授权信息访问,上述两种机制应符合 GB/T 20518—2018 中 5.2.4 的规定。
- c) CA 可选择不指定响应者证书撤销状态的检查方法。在此情况下,可根据 OCSP 请求者的本地安全策略来决定是否对证书进行撤销检查。
- d) 轻量级 OCSP 环境下,在 OCSP 响应程序的证书中宜包含 id-pkix-ocsp-nocheck 扩展,以向请求者表明不需要检查证书状态。另外,在 OCSP 响应者证书中不宜包含授权信息访问和 CRL 分发点扩展。因此,响应者证书应相对较短,并定期更新。见 RFC 5019 中的 2.2.2。

7.3.2.2.4 证书状态发布

OCSP 响应者应按照 GB/T 19714—2005 中 7.3.15 的要求从 CA 获得证书发布的状态。

7.3.2.3 基础响应

基础响应的注解如下。

- a) 基本的响应类型包含:
 - 1) 响应语法的版本,对于本文件的基本响应语法,版本应为 v1(0);
 - 2) 响应者名称或响应者公钥的杂凑值作为 ResponderID 字段值;
 - 3) 响应生成的时间;
 - 4) 对请求中的每个证书的响应;
 - 5) 可选的扩展;
 - 6) 由响应的杂凑值计算得到签名值;
 - 7) 签名算法 OID。
- b) ResponderID 字段信息的用途是允许请求者识别 OCSP 响应者的签名证书,因此,该信息应与签署 OCSP 响应的证书相对应。

- c) 响应者可在 BasicOCSPResponse 结构的 certs 域中包含响应者签名的证书。
- d) 请求中每个证书的响应包括：
 - 1) 提供撤销状态信息的证书的标识(即目标证书)；
 - 2) 证书的撤销状态(在用、已撤销或未知)；如果证书已撤销，则应说明撤销证书的时间，也可说明撤销原因；
 - 3) 响应的有效间隔；
 - 4) 可选扩展。

注：响应为请求中的每个证书包含一个 SingleResponse 结构。

见 RFC 6960 中的 4.2.2.3。

- e) 在轻量级 OCSP 响应中，OCSPResponse 结构应符合以下要求：
 - 1) OCSP 响应仅在 ResponseData.responses 结构中包含一个 SingleResponse 字段；但是，如果需要提高响应预生成性能或缓存效率，预先生成状态响应的 OCSP 响应者可包含附加的 SingleResponse 信息；
 - 2) 响应中不包含 responseExtensions 结构，且请求者忽略响应中无法识别的非关键响应扩展；
 - 3) 若 OCSP 请求中包含响应者无法支持的选项时，响应者尽可能返回最完整的响应，例如，响应者只支持预先生成的响应，并且无法响应包含 Nonce 的 OCSP 请求，则它应返回不包含 Nonce 的响应；
 - 4) 若响应不包含 Nonce 时，请求者应忽略响应 Nonce，并且根据准确的时间确定 OCSP 响应是最新；
 - 5) OCSP 请求中包含响应者无法支持的选项(例如 Nonce)时，响应者可将请求转发给支持的响应者；
 - 6) 响应者可包括 singleResponse.singleExtensions 扩展结构。

见 RFC 5019 中的 2.2.1。

7.4 扩展

7.4.1 概述

本条定义了请求和响应的扩展语法，这些扩展符合 X.509 V3 版本证书的扩展模式，扩展语法符合 GB/T 20518—2018 中 2.4.2 的要求。对请求者和响应者而言，所有扩展均为可选的。对于每个扩展，本条对扩展语法以及需 OCSP 响应者或请求者执行的操作要求进行了定义。关于扩展语法的安全考虑见附录 D。

7.4.2 Nonce

Nonce 是 OCSPRequest 结构或 OCSPResponse 结构中的一个字段，在使用时以加密的方式绑定请求和响应，以防止重放攻击，Nonce 使用时应符合如下要求。

- a) Nonce 的质量符合 GB/T 32915 中的规定。
- b) 在请求中，Nonce 字段作为请求结构中的一个 requestExtension 包含在请求中，而在响应中则作为响应结构中的一个 responseExtension 包含在响应中。在请求和响应中，Nonce 字段将由对象标识符 id-pkix-ocsp-nonce 标识，extnValue 为 Nonce 的值。如果存在 Nonce 扩展，则 Nonce 字段的长度至少为 1 个字节，最多可为 32 个字节。
- c) OCSP 请求中若包含长度为 0 字节或大于 32 字节的 Nonce 扩展，响应者拒绝该请求，并在 OCSPResponseStatus 返回 malformedRequest 的响应结果，见 7.3.1。

- d) 为兼容 GB/T 19713—2005, Nonce 最小长度为 1 字节; OCSP 请求者宜使用长度为 32 字节的 Nonce 扩展。
- e) OCSP 响应者接受长度为 16 字节以上(含 16 字节)的 Nonce 值, 对于小于 16 字节的请求, 选择忽略 Nonce 扩展。

Nonce 定义如下:

```
id-pkix-ocsp          OBJECT IDENTIFIER ::= { id-ad-ocsp }
id-pkix-ocsp-nonce    OBJECT IDENTIFIER ::= { id-pkix-ocsp 2 }
```

Nonce ::= OCTET STRING(SIZE(1..32))

见 RFC 8954 中的 2.1。

7.4.3 CRL 引用

通过 CRL 引用扩展, OCSP 响应者可指出已撤销或已冻结证书所在的 CRL, 该扩展非常适用于当 OCSP 在存储库之间使用或者作为一种复核机制。CRL 可由 URL(CRL 可用的 URL)、序列号(CRL 序列号)或时间点(创建相应的 CRL 的时间点)指定, 且这些扩展为 singleExtensions。CRL 引用扩展的标识符为 id-pkix-ocsp-crl, 值为 CrIID。此项定义如下:

```
id-pkix-ocsp-crl    OBJECT IDENTIFIER ::= { id-pkix-ocsp 3 }
```

```
CrIID ::= SEQUENCE {
    crlUrl      [0] EXPLICIT IA5String OPTIONAL,
    crlNum      [1] EXPLICIT INTEGER OPTIONAL,
    crlTime     [2] EXPLICIT GeneralizedTime OPTIONAL }

```

CrIID 结构的各域含义是:

- 可选项 crlUrl 指定适用于有效 CRL 的 URL, 类型为 IA5String;
- 可选项 crlNum 指定相关 CRL 的 CRL 序列号扩展的值, 类型为 INTEGER;
- 可选项 crlTime 指定发布相应 CRL 的时间点, 类型为 GeneralizedTime。

7.4.4 可接受的响应类型

为指定 OCSP 请求者能处理的响应类型, OCSP 请求者应使用具有 OID 为 id-pkix-ocsp-response 和值为 AcceptableResponses 的可接收的响应类型扩展, 该扩展作为请求中的 requestExtensions 域之一包含在内, AcceptableResponses 域中包含的 OID 是该请求者可接受的各种响应类型的 OID。(例如: id-pkix-ocsp-basic)的 OID。该项定义如下:

```
id-pkix-ocsp-response OBJECT IDENTIFIER ::= { id-pkix-ocsp 4 }
AcceptableResponses ::= SEQUENCE OF OBJECT IDENTIFIER

```

如 7.3.1 所述, OCSP 响应者应能返回 id-pkix-ocsp-basic 类型的响应, 相应的, OCSP 的请求者也能接收和处理 id-pkix-ocsp-basic 类型的响应。

7.4.5 存档截止

响应的 producedAt 时间与间隔保持值之间的差值被定义为证书的“存档截止”时间, 在证书过期后, OCSP 响应者可选择在“存档截止”时间内仍保留相应的撤销信息。

当验证签名所需的数字证书早已过期, OCSP 请求者可使用 OCSP 存档截止时间, 来证明数字签名在生成之日是否有效。

提供历史参考支持的 OCSF 响应者应在响应中包括存档截止扩展,并把存档截止时间作为 OCSF 的 singleExtensions 扩展,存档截止时间由 id-pkix-ocsp-archive-cutoff 和 GeneralizedTime 语法来识别。该项定义如下:

id-pkix-ocsp-archive-cutoff OBJECT IDENTIFIER ::= {id-pkix-ocsp 6}
ArchiveCutoff ::= GeneralizedTime

举例说明,如果响应者采用 7 年保留策略,且 produceAt 值为 t1,那么响应中 ArchiveCutoff 值为 (t1-7 年)。

7.4.6 CRL 条目扩展

CRL 条目扩展应符合 GB/T 20518—2018 中 5.3.4.7 的要求。

7.4.7 服务定位器

OCSF 响应者可以以路由模式运作,即响应者接收到一个请求,并将请求转发至能识别待验证证书的 OCSF 响应者上。本条为上述模式定义了 serviceLocator 请求扩展,该扩展作为一个 singleRequestExtensions 包括在请求中。该项定义如下:

id-pkix-ocsp-service-locator OBJECT IDENTIFIER ::= {id-pkix-ocsp 7}
ServiceLocator ::= SEQUENCE {
 issuer Name,
 locator AuthorityInfoAccessSyntax OPTIONAL}

上述字段的值可从证书的相应字段中获得。

7.4.8 优先使用的签名算法

7.4.8.1 概述

请求者没有选择算法的相关机制,因此存在请求者不支持响应者使用非强制算法签名响应的风险。本条为避免上述风险定义了优先使用的签名算法扩展,请求者可在 OCSF 请求中使用该扩展。见 RFC 6960 中的 4.4.7。

本文件规定 OCSF 响应者应支持国家密码管理部门认可的密码算法的签名响应。

7.4.8.2 扩展语法

请求者可在 OCSF 请求的 requestExtensions 扩展字段中增加优先使用签名算法,该扩展项定义如下:

id-pkix-ocsp-pref-sig-algs OBJECT IDENTIFIER ::= { id-pkix-ocsp 8 }
PreferredSignatureAlgorithms ::= SEQUENCE OF
 PreferredSignatureAlgorithm

PreferredSignatureAlgorithm ::= SEQUENCE {
 sigIdentifier AlgorithmIdentifier,
 pubKeyAlgIdentifier AlgorithmIdentifier OPTIONAL}

sigIdentifier 域指定请求者优先使用的签名算法,例如 Algorithm=SM3WithSM2,签名算法的参数依赖于所选的签名算法,大多数签名算法的参数均为缺省。

pubKeyAlgIdentifier 域指定请求者用于验证 OCSP 响应的公钥密码算法的标识符。pubKeyAlgIdentifier 是可选的,它提供了一种方法来指定公钥密码算法所需的参数,以区分公钥密码算法的不同用法。如果公钥密码算法为 SM2,无参数。

请求者应支持所有指定的优先使用签名算法,并且应按照优先顺序(从最优先到最不优先)指定算法。

7.4.8.3 描述了响应者签名算法的方法,实现对 OCSP 响应进行签名。

见 RFC 6960 中的 4.4.7.1。

7.4.8.3 响应者签名算法选择

7.4.8.3.1 动态响应

所选算法在满足 OCSP 响应者所有安全要求的前提下,响应者可按照以下优先顺序选择支持的签名算法,从而最大限度地确保互操作性,其中第一个选择机制优先级最高:

- a) 选择请求者请求中指定的优先使用的签名算法;
- b) 选择证书颁发者签发证书撤销列表(CRL)所使用的签名算法,证书颁发者为请求中查验证书的证书颁发者;
- c) 选择用于签发 OCSP 请求的签名算法;
- d) 选择已公布为使用带外机制的签名服务的默认签名算法的签名算法;
- e) 选择为使用的 OCSP 版本指定的强制或推荐的签名算法。

响应者应始终应用编号最低的选择机制,从而选择符合响应者加密算法强度标准的已知和支持的算法。

见 RFC 6960 中的 4.4.7.2.1。

7.4.8.3.2 静态响应

为了提高效率,OCSP 响应者在请求之前可生成静态响应。在静态响应生成期间,响应者不能使用请求者请求数据,但可使用历史客户请求作为输入的一部分,来决定使用何种算法来签署预先生成的响应。在选择返回预先生成的响应期间,响应者应使用请求者请求数据。

见 RFC 6960 中的 4.4.7.2.2。

7.4.9 扩展撤销定义

此扩展表示响应者支持“revoked”状态的扩展定义,也适用未签发的证书(见 5.3),其主要目的是允许检查确定响应者的操作类型,使请求者不用解析此扩展来确定响应中证书的状态。

当 OCSP 响应支持未签发证书的“revoked”状态时,则该扩展应包含在 OCSP 响应中;另外,此扩展也可能出现在其他响应中,用来表明响应者实现了扩展的撤销定义。如果响应中包含此扩展,则该扩展应包含在 responseExtensions 字段中,且不能出现在 singleExtensions 字段中。

该扩展由对象标识符 id-pkix-ocsp-extended-revoke 标识。该项定义如下:

id-pkix-ocsp-extended-revoke OBJECT IDENTIFIER ::= {id-pkix-ocsp 9}

该扩展的值应为 NULL,且不应标记为关键扩展。

见 RFC 6960 中的 4.4.8。

附录 A

(规范性)

OCSP 请求和响应的 ASN.1 语法规范

A.1 常规 OCSP ASN.1 语法规范

OCSP 请求和响应的 ASN.1 语法规范定义如下：

DEFINITIONS EXPLICIT TAGS ::=

BEGIN

IMPORTS

Extensions{ }, EXTENSION, ATTRIBUTE

FROM PKIX-CommonTypes-2009 -- 引用 [RFC5912], 应符合 GB/T 20518—2018 中的语法要求

{iso(1) identified-organization(3) dod(6) internet(1) security(5)

mechanisms(5) pkix(7) id-mod(0) id-mod-pkixCommon-02(57)}

AlgorithmIdentifier{ }, DIGEST-ALGORITHM, SIGNATURE-ALGORITHM, PUBLIC-KEY

FROM AlgorithmInformation-2009 -- 引用 [RFC5912], 应符合 GB/T 20518—2018 中的语法要求

--SM3WithSM2 和 SM3 见 GB/T 33560-2017

{iso(1) identified-organization(3) dod(6) internet(1) security(5)

mechanisms(5) pkix(7) id-mod(0)

id-mod-algorithmInformation-02(58)}

AuthorityInfoAccessSyntax, GeneralName, CrlEntryExtensions

FROM PKIX1Implicit-2009 -- 引用 [RFC5912], 应符合 GB/T 20518—2018 中的语法要求

{iso(1) identified-organization(3) dod(6) internet(1) security(5)

mechanisms(5) pkix(7) id-mod(0) id-mod-pkix1-implicit-02(59)}

Name, CertificateSerialNumber, id-kp, id-ad-ocsp, Certificate

FROM PKIX1Explicit-2009 -- 引用 [RFC5912], 应符合 GB/T 20518—2018 中的语法要求

{iso(1) identified-organization(3) dod(6) internet(1) security(5)

mechanisms(5) pkix(7) id-mod(0) id-mod-pkix1-explicit-02(51)}

OCSPRequest ::= SEQUENCE {

tbsRequest TBSRequest,

optionalSignature [0] EXPLICIT Signature OPTIONAL }

TBSRequest ::= SEQUENCE {

version [0] EXPLICIT Version DEFAULT v1,

requestorName [1] EXPLICIT GeneralName OPTIONAL,

requestList SEQUENCE OF Request,

```

requestExtensions      [2]      EXPLICIT Extensions { {re-ocsp-nonce |
                                re-ocsp-response, ...,
                                re-ocsp-preferred-signature-algorithms} } OPTIONAL }

Signature      ::=      SEQUENCE {
    signatureAlgorithm      AlgorithmIdentifier
                            { SIGNATURE-ALGORITHM, {...} },
    signature               BIT STRING,
    certs                  [0]      EXPLICIT SEQUENCE OF Certificate OPTIONAL }

Version  ::=  INTEGER  {  v1(0)  }

Request ::=      SEQUENCE {
    reqCert              CertID,
    singleRequestExtensions      [0]      EXPLICIT Extensions
                                    { {re-ocsp-service-locator,
                                    ...} } OPTIONAL }

CertID ::= SEQUENCE {
    hashAlgorithm      AlgorithmIdentifier
                            {DIGEST-ALGORITHM, {...} },
    issuerNameHash      OCTET STRING, -- 颁发者证书主题(DN)的杂凑值
    issuerKeyHash        OCTET STRING, -- 颁发者证书公钥的杂凑值
    serialNumber         CertificateSerialNumber }

OCSPResponse ::= SEQUENCE {
    responseStatus      OCSPResponseStatus,
    responseBytes        [0]      EXPLICIT ResponseBytes OPTIONAL }

OCSPResponseStatus ::= ENUMERATED {
    successful              (0), -- 响应被有效确认
    malformedRequest        (1), -- 请求格式错误
    internalError           (2), -- 内部错误
    tryLater                (3), -- 稍候重试
                            -- (4) 未使用
    sigRequired             (5), -- 应对请求签名
    unauthorized            (6) -- 请求未被授权
}

RESPONSE ::= TYPE-IDENTIFIER

ResponseSet RESPONSE ::= { basicResponse, ... }

```

ResponseBytes ::= SEQUENCE{
 responseType RESPONSE
 &.id({ResponseSet}),
 Response OCTET STRING(CONTAINING RESPONSE.
 &.Type({ResponseSet}{@responseType}))}

basicResponse RESPONSE ::=
 {BasicOCSPResponse IDENTIFIED BY id-pkix-ocsp-basic}

BasicOCSPResponse ::= SEQUENCE{
 tbsResponseData ResponseData,
 signatureAlgorithm AlgorithmIdentifier{SIGNATURE-ALGORITHM,
 {sa-rsaWithSHA256|SM2WithSM3}},
 signature BIT STRING,
 certs [0] EXPLICIT SEQUENCE OF Certificate OPTIONAL}

ResponseData ::= SEQUENCE{
 version [0] EXPLICIT Version DEFAULT v1,
 responderID ResponderID
 producedAt GeneralizedTime,
 responses SEQUENCE OF SingleResponse
 responseExtensions [1] EXPLICIT Extensions
 {{re-ocsp-nonce,...
 re-ocsp-extended-revoke}} OPTIONAL}

ResponderID ::= CHOICE{
 byName [1] Name,
 byKey [2] KeyHash}

KeyHash ::= OCTET STRING -- hash of responder's public key

SingleResponse ::= SEQUENCE{
 certID CertID
 certStatus CertStatus
 thisUpdate GeneralizedTime
 nextUpdate [0] EXPLICIT GeneralizedTime OPTIONAL
 singleExtensions [1] EXPLICIT Extensions {{re-ocsp-crl|
 re-ocsp-archive-cutoff|
 CrlEntryExtensions,...}} OPTIONAL}

CertStatus ::= CHOICE{
 good [0] IMPLICIT NULL,
 revoked [1] IMPLICIT RevokedInfo,
 unknown [2] IMPLICIT UnknownInfo}

RevokedInfo ::= SEQUENCE{
 revocationTime GeneralizedTime,
 revocationReason [0] EXPLICIT CRLReason OPTIONAL}

UnknownInfo ::= NULL

ArchiveCutoff ::= GeneralizedTime

AcceptableResponses ::= SEQUENCE OF RESPONSE.&id({ResponseSet})

ServiceLocator ::= SEQUENCE{
 issuer Name,
 locator AuthorityInfoAccessSyntax}

CrlID ::= SEQUENCE{
 crlUrl [0] EXPLICIT IA5String OPTIONAL
 crlNum [1] EXPLICIT INTEGER OPTIONAL
 crlTime [2] EXPLICIT GeneralizedTime OPTIONAL}

PreferredSignatureAlgorithms ::= SEQUENCE OF PreferredSignatureAlgorithm

PreferredSignatureAlgorithms ::= SEQUENCE{
 sigIdentifier AlgorithmIdentifier{SIGNATURE-ALGORITHM, {...}}
 pubKeyAlgIdentifier AlgorithmIdentifier{PUBLIC-KEY, {...}} OPTIONAL
 }

--Certificate Extensions

ext-ocsp-nocheck EXTENSION ::= {SYNTAX NULL IDENTIFIED
 BY id-pkix-ocsp-nocheck}

--Request Extensions

re-ocsp-nonce EXTENSION ::= {SYNTAX OCTET STRING(SIZE(1...32)) IDENTIFIED
 BY id-pkix-ocsp-nonce}

re-ocsp-response EXTENSION ::= {SYNTAX AcceptableResponse IDENTIFIED
 BY id-pkix-ocsp-response}

re-ocsp-service-locator EXTENSION ::= {SYNTAX ServiceLocator
 IDENTIFIED BY
 id-pkix-ocsp-service-locator}

re-ocsp-preferred-signature-algorithms EXTENSION ::= {
 SYNTAX PreferredSignatureAlgorithms
 IDENTIFIED BY id-pkix-ocsp-pref-sig-algs}

--Response Extensions

re-ocsp-crl EXTENSION ::= { SYNTAX CrlID IDENTIFIED BY
id-pkix-ocsp-crl }
re-ocsp-archive-cutoff EXTENSION ::= { SYNTAX ArchiveCutoff IDENTIFIED BY
id-pkix-ocsp-archive-cutoff }
re-ocsp-extended-revoke EXTENSION ::= { SYNTAX NULL IDENTIFIED BY
id-pkix-ocsp-extended-revoke }

--ObjectIdentifiers

id-kp-OCSPSigning	OBJECT	IDENTIFIER ::= { id-kp 9 }
id-pkix-ocsp	OBJECT	IDENTIFIER ::= id-ad-ocsp
id-pkix-ocsp-basic	OBJECT	IDENTIFIER ::= { id-pkix-ocsp 1 }
id-pkix-ocsp-nonce	OBJECT	IDENTIFIER ::= { id-pkix-ocsp 2 }
id-pkix-ocsp-crl	OBJECT	IDENTIFIER ::= { id-pkix-ocsp 3 }
id-pkix-ocsp-response	OBJECT	IDENTIFIER ::= { id-pkix-ocsp 4 }
id-pkix-ocsp-nocheck	OBJECT	IDENTIFIER ::= { id-pkix-ocsp 5 }
id-pkix-ocsp-archive-cutoff	OBJECT	IDENTIFIER ::= { id-pkix-ocsp 6 }
id-pkix-ocsp-service-locator	OBJECT	IDENTIFIER ::= { id-pkix-ocsp 7 }
id-pkix-ocsp-pref-sig-algs	OBJECT	IDENTIFIER ::= { id-pkix-ocsp 8 }
id-pkix-ocsp-extended-revoke	OBJECT	IDENTIFIER ::= { id-pkix-ocsp 9 }

END

见 RFC 6960 中的 B.2。

A.2 轻量级 OCSP ASN.1 语法规范

轻量级 OCSP 请求和响应的 ASN.1 语法规范定义如下：

DEFINITIONS EXPLICIT TAGS ::=

BEGIN

IMPORTS

Extensions{ }, EXTENSION, ATTRIBUTE

FROM PKIX-CommonTypes-2009 -- 引用 [RFC5912],应符合 GB/T 20518—2018 中的语法要求
{iso(1) identified-organization(3) dod(6) internet(1) security(5)
mechanisms(5) pkix(7) id-mod(0) id-mod-pkixCommon-02(57)}

AlgorithmIdentifier{ }, DIGEST-ALGORITHM, SIGNATURE-ALGORITHM, PUBLIC-KEY
FROM AlgorithmInformation-2009 -- 引用 [RFC5912],应符合 GB/T 20518—2018 中的语法要求
--SM3WithSM2 和 SM3 见 GB/T 33560-2017
{iso(1) identified-organization(3) dod(6) internet(1) security(5)
mechanisms(5) pkix(7) id-mod(0)
id-mod-algorithmInformation-02(58)}

AuthorityInfoAccessSyntax, GeneralName, CrlEntryExtensions

FROM PKIX1Implicit-2009 -- 引用 [RFC5912],应符合 GB/T 20518—2018 中的语法要求
 {iso(1) identified-organization(3) dod(6) internet(1) security(5)
 mechanisms(5) pkix(7) id-mod(0) id-mod-pkix1-implicit-02(59)}

Name, CertificateSerialNumber, id-kp, id-ad-ocsp, Certificate

FROM PKIX1Explicit-2009 -- 引用 [RFC5912],应符合 GB/T 20518—2018 中的语法要求
 {iso(1) identified-organization(3) dod(6) internet(1) security(5)
 mechanisms(5) pkix(7) id-mod(0) id-mod-pkix1-explicit-02(51)}

OCSPPRequest ::= SEQUENCE {
 tbsRequest TBSRequest,
 optionalSignature [0] EXPLICIT Signature OPTIONAL }

TBSRequest ::= SEQUENCE {
 version [0] EXPLICIT Version DEFAULT v1,
 requestorName [1] EXPLICIT GeneralName OPTIONAL,
 requestList SEQUENCE OF Request,
 requestExtensions [2] EXPLICIT Extensions {{re-ocsp-nonce}} OPTIONAL }

Version ::= INTEGER { v1(0) }

Request ::= SEQUENCE {
 reqCert CertID,
 }

CertID ::= SEQUENCE {
 hashAlgorithm AlgorithmIdentifier
 {DIGEST-ALGORITHM, {...}},
 issuerNameHash OCTET STRING, -- 颁发者证书主题(DN)的杂凑值
 issuerKeyHash OCTET STRING, -- 颁发者证书公钥的杂凑值
 serialNumber CertificateSerialNumber }

OCSPPResponse ::= SEQUENCE {
 responseStatus OCSPPResponseStatus,
 responseBytes [0] EXPLICIT ResponseBytes OPTIONAL }

OCSPPResponseStatus ::= ENUMERATED {
 successful (0), -- 响应被有效确认
 malformedRequest (1), -- 请求格式错误
 internalError (2), -- 内部错误
 tryLater (3), -- 稍候重试

```

-- (4) 未使用
sigRequired      (5), -- 应对请求签名
unauthorized      (6) -- 请求未被授权
}

RESPONSE ::= TYPE-IDENTIFIER

ResponseSet RESPONSE ::= { basicResponse, ... }

ResponseBytes ::=
    SEQUENCE {
        responseType      RESPONSE
        &.id({ResponseSet}),
        Response          OCTET STRING (CONTAINING RESPONSE.
        &.Type({ResponseSet}{@responseType})) }

basicResponse RESPONSE ::=
    { BasicOCSPResponse IDENTIFIED BY id-pkix-ocsp-basic }

BasicOCSPResponse ::= SEQUENCE {
    tbsResponseData      ResponseData,
    signatureAlgorithm    AlgorithmIdentifier { SIGNATURE-ALGORITHM,
        { sa-rsaWithSHA256 | SM2WithSM3 } },
    signature            BIT STRING,
    certs                [0] EXPLICIT SEQUENCE OF Certificate }

ResponseData ::= SEQUENCE {
    version              [0] EXPLICIT Version DEFAULT v1,
    responderID          ResponderID
    producedAt           GeneralizedTime,
    responses            SEQUENCE OF SingleResponse
}

ResponderID ::= CHOICE {
    byName [1] Name,
    byKey  [2] KeyHash }

KeyHash ::= OCTET STRING -- hash of responder's public key

SingleResponse ::= SEQUENCE {
    certID              CertID
    certStatus          CertStatus
    thisUpdate          GeneralizedTime
    nextUpdate          [0] EXPLICIT GeneralizedTime
    singleExtensions    [1] EXPLICIT Extensions { { re-ocsp-crl |

```

re-ocsp-archive-cutoff|
 CrlEntryExtensions, ... } } OPTIONAL }

CertStatus ::= CHOICE{
 good [0] IMPLICIT NULL,
 revoked [1] IMPLICIT RevokedInfo,
 unknown [2] IMPLICIT UnknownInfo }

RevokedInfo ::= SEQUENCE{
 revocationTime GeneralizedTime,
 revocationReason [0] EXPLICIT CRLReason OPTIONAL }

UnknownInfo ::= NULL

ArchiveCutoff ::= GeneralizedTime

AcceptableResponses ::= SEQUENCE OF RESPONSE.&id({ResponseSet})

ServiceLocator ::= SEQUENCE{
 issuer Name,
 locator AuthorityInfoAccessSyntax }

CrlID ::= SEQUENCE{
 crlUrl [0] EXPLICIT IA5String OPTIONAL
 crlNum [1] EXPLICIT INTEGER OPTIONAL
 crlTime [2] EXPLICIT GeneralizedTime OPTIONAL }

PreferredSignatureAlgorithms ::= SEQUENCE OF PreferredSignatureAlgorithm

PreferredSignatureAlgorithms ::= SEQUENCE{
 sigIdentifier AlgorithmIdentifier{SIGNATURE-ALGORITHM, { ... } }
 pubKeyAlgIdentifier AlgorithmIdentifier{PUBLIC-KEY, { ... } } OPTIONAL
 }

--Certificate Extensions

ext-ocsp-nocheck EXTENSION ::= {SYNTAX NULL IDENTIFIED
 BY id-pkix-ocsp-nocheck }

--equest Extensions

re-ocsp-nonce EXTENSION ::= {SYNTAX OCTET STRING(SIZE(1...32)) IDENTIFIED
 BY id-pkix-ocsp-nonce }

re-ocsp-response EXTENSION ::= {SYNTAX AcceptableResponse IDENTIFIED

BY id-pkix-ocsp-response}

re-ocsp-service-locator EXTENSION ::= { SYNTAX ServiceLocator
IDENTIFIED BY
id-pkix-ocsp-service-locator }

re-ocsp-preferred-signature-algorithms EXTENSION ::= {
SYNTAX PreferredSignatureAlgorithms
IDENTIFIED BY id-pkix-ocsp-pref-sig-algs }

--Response Extensions

re-ocsp-crl EXTENSION ::= { SYNTAX CrlID IDENTIFIED BY
id-pkix-ocsp-crl }

re-ocsp-archive-cutoff EXTENSION ::= { SYNTAX ArchiveCutoff IDENTIFIED BY
id-pkix-ocsp-archive-cutoff }

--ObjectIdentifiers

id-kp-OCSPSigning	OBJECT	IDENTIFIER ::= { id-kp 9 }
id-pkix-ocsp	OBJECT	IDENTIFIER ::= id-ad-ocsp
id-pkix-ocsp-basic	OBJECT	IDENTIFIER ::= { id-pkix-ocsp 1 }
id-pkix-ocsp-nonce	OBJECT	IDENTIFIER ::= { id-pkix-ocsp 2 }
id-pkix-ocsp-crl	OBJECT	IDENTIFIER ::= { id-pkix-ocsp 3 }
id-pkix-ocsp-response	OBJECT	IDENTIFIER ::= { id-pkix-ocsp 4 }
id-pkix-ocsp-nocheck	OBJECT	IDENTIFIER ::= { id-pkix-ocsp 5 }
id-pkix-ocsp-archive-cutoff	OBJECT	IDENTIFIER ::= { id-pkix-ocsp 6 }
id-pkix-ocsp-service-locator	OBJECT	IDENTIFIER ::= { id-pkix-ocsp 7 }
id-pkix-ocsp-pref-sig-algs	OBJECT	IDENTIFIER ::= { id-pkix-ocsp 8 }

END

附 录 B

(规范性)

基于 HTTP 的 OCSP 请求和响应

B.1 常规 OCSP 请求与响应的构造

B.1.1 请求

基于 HTTP 的 OCSP 请求使用 HTTP 协议中资源获取方法(GET)或 HTTP 协议中利用资源动作方法(POST)来提交。在启用 HTTP 缓存时,可通过 GET 方法提交较小请求(编码后小于或等于 255 字节),对于不需要 HTTP 缓存或大于 255 个字节的请求,则应通过 POST 方法提交。

使用 GET 方法的 OCSP 请求构造如下:

GET{url}/{DER 编码的 OCSPRequest 二进制值进行使用可打印字符编码(Base64)后再进行 url-encoding 编码}

其中{url}从正在查询状态的证书的机构信息访问扩展的值或者 OCSP 请求者的本地配置获得。

使用 POST 方法的 OCSP 请求构造如下:

HTTP 请求头中 Content-Type 属性的值为“application/ocsp-request”,而消息正文是 OCSPRequest 的 DER 编码的二进制值。

B.1.2 响应

基于 HTTP 的 OCSP 响应构造如下:

HTTP 响应头中 Content-Type 属性的值为“application/ocsp-response”,Content-Length 属性指定响应的长度,而消息正文是 OCSPResponse 的 DER 编码的二进制值。其他的不能被客户端识别的 HTTP 头可能存在于响应中,可被忽略。

B.2 轻量级 OCSP 请求与响应的构造

B.2.1 概述

高价值的电子交易或高敏感度信息和操作需要 OCSP 响应者为每个证书提供及时和安全的状态查询服务,OCSP 通常部署在带宽和处理能力充足不受限制的环境下,但随着 PKI 技术使用的不断发展,OCSP 的应用的场景也不断增加,例如在大规模的 PKI 环境中,从效率和成本角度出发,常规 OCSP 的应用会受到限制。

为了适应在非常大规模(高容量)PKI 环境或需要轻量级解决方案以最大限度地减少带宽和请求者/响应者处理能力的 PKI 环境下的应用,轻量级 OCSP 在常规 OCSP 的基础上,规范了 OCSP 请求者和响应者允许采用的配置和操作,可采用的操作如下:

- a) 预生成 OCSP 响应;
- b) 减小 OCSP 消息大小以降低带宽使用;
- c) 响应消息缓存在请求者和响应者。

B.2.2 请求

轻量级 OCSP 要求响应者应支持基于 HTTP 的请求和响应。当发送总共小于或等于 255 字节(编码后)的请求时,请求者应使用 GET 方法(以启用 OCSP 响应缓存);当发送大于 255 字节的 OCSP 请求时,则应使用 POST 方法提交。

在构造 GET 消息时,OCSP 请求者应对 OCSPRequest 结构进行 Base64 编码,并将其附加到机构信息访问扩展中 id-ad-ocsp 指定的 URI。请求者不得在 Base64 编码的字符串中包含 CR 或 LF 字符,而且应对 Base64 编码的 OCSPRequest 结构进行 url-encoding。例如:

http://ocsp.example.com/MEowsDVGMEYavkyaKBgggnkiG9v0TsVYAGG7spbGTKpl2dAdeJaV267ovkYakinESVKD0mJeBarSzhv%2FVVIKLZhs%2Fg9xF8ooSizol80Mbp%ZD%ZD

B.2.3 响应

基于 HTTP 的 OCSP 响应构造应符合如下规定,具体内容如下:

为了支持可缓存的响应(即包含 nextUpdate 值的响应),响应的消息正文中需包含 OCSPResponse 的 DER 编码的二进制值。响应前应带有以下 HTTP 报头:

```
content-type: application/ocsp-response
content-length: <OCSP response length>
last-modified: <producedAt [HTTP] date>
ETag: "<strong validator>"
expires: <nextUpdate [HTTP] date>
cache-control: max-age=<n>, public, no-transform, must-revalidate
date: <current [HTTP] date>
```



附录 C

(资料性)

OCSP 请求和响应 ASN.1 语法消息示例

C.1 常规 OCSP ASN.1 消息示例

C.1.1 请求

```

SEQUENCE {
    SEQUENCE {
        [0] { INTEGER 0 }
        SEQUENCE {
            SEQUENCE {
                SEQUENCE {
                    SEQUENCE {
                        SEQUENCE {
                            OBJECT IDENTIFIER  SM3 (1.2.156.197.1.401)
                            NULL
                        }
                    }
                    OCTET STRING
                        42 0E 5E 17 FC 60 D8 05 D9 EF E2 53 E7 AB 7F 3A
                        A8 84 AA 5E 46 5B E7 B8 1F C6 B1 35 AD FF D1 CC
                    OCTET STRING
                        D2 64 17 79 CB 1E 1C 1C 0C 6E 28 56 B5 16 4A 4A
                        00 1A C1 B0 74 D7 B4 55 9D 2A 99 1F 0E 4A E3 5F
                    INTEGER
                        09 34 23 72 E2 3A EF 46 7C 83 2D 07 F8 DC 22 BA
                }
            }
        }
    }
}

[0]{
    SEQUENCE {
        SEQUENCE {
            OBJECT IDENTIFIER  SM3WithSM2 (1.2.156.10197.1.501)
        }
        BIT STRING{
            SEQUENCE {
                INTEGER
                    67 0C 3E 99 24 5E 0D 1C 06 B7 47 DE B3 12 4D C8
                    43 BB 8B A6 1F 03 5A 7D 09 38 25 1F 5D D4 CB FC
                INTEGER
                    00 96 F5 45 3B 13 0D 89 0A 1C DB AE 32 20 9A 50 EE
            }
        }
    }
}

```


40 78 36 FD 12 49 32 F6 9E 7D 49 DC AD 4F 14 F2

}

}

}

}

}

C.1.2 响应

SEQUENCE {

ENUMERATED 0

[0] {

SEQUENCE {

OBJECT IDENTIFIER ocsBasic (1.3.6.1.5.5.7.48.1.1)

OCTET STRING, response

{

SEQUENCE{

[0] { INTEGER 0 }

[1] {

SEQUENCE {

SET {

SEQUENCE {

OBJECT IDENTIFIER commonName(2.5.4.3)

UTF8String Example ‘OCSP Responder’

}

}

SET {

SEQUENCE {

OBJECT IDENTIFIER organizationName (2.5.4.10)

UTF8String ‘组织名称’

}

}

SET {

SEQUENCE {

OBJECT IDENTIFIER organizationalUnitName (2.5.4.11)

UTF8String ‘部门名称’

}

}

SET {

SEQUENCE {

OBJECT IDENTIFIER countryName(2.5.4.6)

UTF8String ‘CN’

}

}

```

    }
    GeneralizedTime '20211108232625Z'
    SEQUENCE {
        SEQUENCE {
            SEQUENCE {
                SEQUENCE {
                    OBJECT IDENTIFIER SM3 (1.2.156.197.1.401)
                    NULL
                }
            }
            OCTET STRING
                87 0C 3E 99 24 5E 0D 1C 06 B7 47 DE B3 12 4D C8
                43 BB 8B A6 1F 03 5A 7D 09 38 25 1F 5D D4 CB FC
            OCTET STRING
                2D DC 8E 66 83 EF 57 49 61 FF 69 8F 61 CD D1 1E
                9D 9C 16 72 72 E6 1D F0 84 4F 4A 77 02 D7 E8 39
            INTEGER
                09 34 23 72 E2 3A EF 46 7C 83 2D 07 F8 DC 22 BA
        }
        CHOICE [0]
        GeneralizedTime '20211108232625Z'
        [0] {
            GeneralizedTime '20211108231823Z'
        }
    }
}

SEQUENCE {
    OBJECT IDENTIFIER SM3WithSM2 (1.2.156.10197.1.501)
}

BIT STRING {
    SEQUENCE {
        INTEGER
            17 0C 3E 99 24 5E 0D 1C 06 B7 47 DE B3 12 4D C8
            43 BB 8B A6 1F 03 5A 7D 09 38 25 1F 5D D4 CB FC
        INTEGER
            00 96 F5 45 3B 13 0D 89 0A 1C DB AE 32 20 9A 50 EE
            40 78 36 FD 12 49 32 F6 9E 7D 49 DC AD 4F 14 F2
    }
}

[0] {
    SEQUENCE {
        SEQUENCE {
            SEQUENCE {
                [0] {

```

```

    INTEGER 2
  }
  INTEGER
    49 4A 02 37 1B 1E 70 67 41 6C 9F 06 2F D8 FE DA
  SEQUENCE {
    OBJECT IDENTIFIER SM3WithSM2 (1.2.156.10197.1.501)
  }
  SEQUENCE {
    SET {
      SEQUENCE {
        OBJECT IDENTIFIER commonName (2.5.4.3)
        UTF8String 'ExampleCA SM2 CA'
      }
    }
    SET {
      SEQUENCE {
        OBJECT IDENTIFIER countryName(2.5.4.6)
        UTF8String 'CN'
      }
    }
    SEQUENCE {
      GeneralizedTime '20210810000000Z'
      GeneralizedTime '20300610235959Z'
    }
    SEQUENCE {
      SET {
        SEQUENCE {
          OBJECT IDENTIFIER commonName (2.5.4.3)
          UTF8String 'Example OCSP Responder'
        }
      }
      SET {
        SEQUENCE {
          OBJECT IDENTIFIER organizationName (2.5.4.10)
          UTF8String '组织名称'
        }
      }
      SET {
        SEQUENCE {
          OBJECT IDENTIFIER organizationalUnitName (2.5.4.11)
          UTF8String '部门名称'
        }
      }
    }
  }
  SET {

```

```

SEQUENCE {
    OBJECT IDENTIFIER countryName(2.5.4.6)
    UTF8String 'CN'
}
}
SEQUENCE {
    SEQUENCE {
        OBJECT IDENTIFIER SM2 (1.2.156.197.1.301)
        NULL
    }
    BIT STRING {
        04 47 ec d9 54 1b e1 13 67 5f 62 7b 59 5f 8a e7 c1 b2 23 e3 71 7a 33 d3
        49 98 2f 3e 88 93 b2 de ef a2 34 75 5a 25 1d 70 49 28 c7 10 66 e5 0b 9c c8
        09 57 81 ae 41 ac 3a c1 79 a0 99 32 4d b2 11 65
    }
}
[3] {
    SEQUENCE {
        SEQUENCE {
            OBJECT IDENTIFIER basicConstraints (2.5.29.19)
            OCTET STRING, encapsulates {
                SEQUENCE {}
            }
        }
        SEQUENCE {
            OBJECT IDENTIFIER extKeyUsage (2.5.29.37)
            OCTET STRING, encapsulates {
                SEQUENCE {
                    OBJECT IDENTIFIER ocspSigning (1.3.6.1.5.5.7.3.9)
                }
            }
        }
    }
    SEQUENCE {
        OBJECT IDENTIFIER keyUsage (2.5.29.15)
        OCTET STRING, encapsulates {
            BIT STRING 7 unused bits
            '1' B (bit 0)
        }
    }
    SEQUENCE {
        OBJECT IDENTIFIER ocspNoCheck (1.3.6.1.5.5.7.48.1.5)
        OCTET STRING, encapsulates {
            NULL

```

```
    }
  }
}
}
SEQUENCE {
  OBJECT IDENTIFIER SM3WithSM2 (1.2.156.10197.1.501)
}
BIT STRING{
  SEQUENCE
  {
    INTEGER
      06 03 55 1D 13 04 05 30 03 01 01 FF 30 0B 06 03
      55 1D 0F 04 04 03 02 02 E4 30 0A 06 08 2A 81 1C
    INTEGER
      12 59 5A E9 FA C8 C9 87 4A 1D 9F CE 1E FF F0 71
      46 F6 5C 58 23 31 A8 46 44 D2 A3 1D 30 1B 30 0C
  }
}
}
}
}
}
}
}
```

C.2 轻量级 OCSF 消息示例

C.2.1 请求

```
SEQUENCE {
  SEQUENCE {
    SEQUENCE {
      SEQUENCE {
        SEQUENCE {
          SEQUENCE {
            OBJECT IDENTIFIER SM3 (1.2.156.197.1.401)
            NULL
          }
        }
      }
    }
  }
  OCTET STRING
    42 0E 5E 17 FC 60 D8 05 D9 EF E2 53 E7 AB 7F 3A
    A8 84 AA 5E 46 5B E7 B8 1F C6 B1 35 AD FF D1 CC
  OCTET STRING
    D2 64 17 79 CB 1E 1C 1C 0C 6E 28 56 B5 16 4A 4A
```

```

00 1A C1 B0 74 D7 B4 55 9D 2A 99 1F 0E 4A E3 5F
INTEGER
09 34 23 72 E2 3A EF 46 7C 83 2D 07 F8 DC 22 BA
}
}
}
}
}
}

```

C.2.2 响应

```

SEQUENCE {
  ENUMERATED 0
  [0] {
    SEQUENCE {
      OBJECT IDENTIFIER ocsBasic (1.3.6.1.5.5.7.48.1.1)
      OCTET STRING, encapsulates {
        SEQUENCE {
          SEQUENCE {
            [0] {
              INTEGER 0
            }
            [2] {
              OCTET STRING
                29 23 BE 84 E1 6C D6 AE 52 90 49 F1 F1 BB E9 EB
                B3 A6 DB 3C
            }
            GeneralizedTime '2021110735244Z'
          }
          SEQUENCE {
            SEQUENCE {
              SEQUENCE {
                OBJECT IDENTIFIER SM3 (1.2.156.197.1.401)
                NULL
              }
              OCTET STRING
                8E 19 DB BA 1E 1E 72 FF 32 F4 44 E0 B2 77 1C D7
                3C 9E 78 F3 D1 82 68 86 63 12 7F A4 6F F0 4D 84
              OCTET STRING
                42 0E 5E 17 FC 60 D8 05 D9 EF E2 53 E7 AB 7F 3A
                A8 84 AA 5E 46 5B E7 B8 1F C6 B1 35 AD FF D1 CC
              INTEGER
                09 34 23 72 E2 3A EF 46 7C 83 2D 07 F8 DC 22 BA
            }
          }
        }
      }
    }
  }
}

```

```
CHOICE [0]
  GeneralizedTime '20211107235244Z'
  [0] {
    GeneralizedTime '20211108235844Z'
  }
}
}
}
}
SEQUENCE {
  OBJECT IDENTIFIER SM3WithSM2 (1.2.156.10197.1.501)
}
BIT STRING{
  SEQUENCE {
    INTEGER
    00 F4 60 58 A4 F1 F5 18 96 1D 4A B2 DD 0A 06 5C F3
    C2 2E 0A 0F 97 68 D2 16 DF 44 30 49 DC 30 C6 EA
    INTEGER
    00 D7 C7 CD F7 6A FE 82 EF 49 A6 59 60 85 FE FD 1B
    D4 78 F0 FC 4B 6C E1 96 70 F1 16 4A 02 21 00 D9
  }
}
}
}
}
}
}
}
```

附录 D

(资料性)

安全考虑

D.1 概述

为使服务有效,查验证书状态的应用程序和提供证书状态查询的响应者需要通过网络相连接,在网络中可能会遇到拒绝服务攻击、重放攻击、中间人攻击、随机数冲突、缓存机制安全、证书序列号冲突等问题,本附录针对这些问题提供如下解决方案。

D.2 优先使用的签名算法

D.2.1 签名算法的选择

响应签名算法的选择机制应足够安全,并且能确保目标应用程序不会受到密码分析的攻击。

在大多数应用程序中,签名算法至少与签署被查询原始证书状态所使用的签名算法一样安全。但不适用于签名算法已经被认为是不可信赖的情况。

见 RFC 6960 中的 5.1。

D.2.2 不安全算法的使用

响应者不是每次都能生成请求者支持且符合相关密码国家标准和行业标准的签名响应,在这种情况下,OCSP 响应者运营商应平衡缺陷安全解决方案存在的风险和强制升级的成本,包括最终用户选择较低安全性或没有安全性的替代方案。

在归档应用程序中,OCSP 响应者会被要求报告某个历史日期证书的有效性,但这样的证书可能会使用被认为是不安全的签名方法。在上述情况下,响应者应选择安全的签名机制来生成签名。

请求者应在响应中接受它在请求中所指定的优先签名算法,因此,请求者不应将任何不受支持或被认为是不安全的算法指定为优先签名算法。

见 RFC 6960 中的 5.1.1。

D.2.3 中间人降级攻击

请求者指定优先签名算法的机制并不能防止中间人降级攻击,这个限制不被认为是一个重要的安全问题,因为即使请求者进行了请求,OCSP 响应者也不应使用弱算法对 OCSP 响应进行签名,此外,无论使用哪种机制来确定响应的签名算法,客户端都可拒绝不符合自身的可接受加密安全性标准的 OCSP 响应。

见 RFC 6960 中的 5.1.2。

D.2.4 拒绝服务攻击

本文件的算法灵活性机制增加了拒绝服务攻击面,在这种攻击中,客户端请求被更改为响应者不支持的算法。响应者如果检测到可疑的行为,可限制来自特定 IP 地址的传入请求的速率。

D.3 重放攻击

Nonce 扩展用于避免重放攻击,因为即使客户端在请求中发送了 Nonce 扩展,OCSP 响应者在响应中也可能选择不发送 Nonce 扩展,所以路径攻击者可拦截 OCSP 请求,并在没有 Nonce 扩展的情况下

使用响应者中较早的响应进行响应,针对上述问题,可通过在 OCSP 响应中的 thisUpdate 和 nextUpdate 字段之间使用较短的时间间隔配置响应者来缓解此问题。

D.4 一次性随机数冲突

如果请求者在 OCSP 请求中使用的 Nonce 的值是可预测的,则攻击者可能使用预测的随机数预取响应并重新播放随机数,从而破坏了使用随机数的目的,所以,OCSP 请求中的随机数扩展的值需要使用符合随机数标准的随机数,并且是在创建 OCSP 请求时新生成的。此外,如果随机数的长度太小(例如,1 个字节),则路径上的攻击者可预取所有可能随机数值的响应,并重新播放匹配的随机数。

见 RFC 8954 中的 3.2。

D.5 缓存机制安全

如果中间级响应者没有正确配置,或缓存管理出错,在某些场景中依赖 HTTP 高速缓存,可能导致一些意外的结果。在部署基于 HTTP 的 OCSP 服务时,本文件建议实施者需考虑 HTTP 缓存机制的可靠性。

D.6 证书序列号冲突

如果请求者能够预测或猜测将要签发证书的序列号,则对从未签发过的证书以“revoked”状态进行响应时,可能会使某些请求者获得尚未签发但即将签发的证书的撤销响应,这样的预测对于使用顺序证书序列号来分配签发证书的 CA 来说是很容易的。文件中通过使用冻结原因代码达到兼容要求来处理此风险,从而避免永久撤销序列号。对于支持对未签发证书的状态请求进行“revoked”响应的 CA,采用分配具有高熵的随机证书序列号值的方法可完全避免此问题。

见 RFC 6960 中的第 5 章。



参 考 文 献

- [1] GM/T 0014—2012 数字证书认证系统密码协议规范
- [2] RFC 2616 Hypertext Transfer Protocol—HTTP/1.1
- [3] RFC 5019 The Lightweight Online Certificate Status Protocol (OCSP) Profile for High—
Volume Environments
- [4] RFC 5912 New ASN.1 Modules for the Public Key Infrastructure Using X.509 (PKIX)
- [5] RFC 6960 X.509 Internet Public Key Infrastructure Online Certificate Status Protocol—
OCSP
- [6] RFC 8954 Online Certificate Status Protocol (OCSP) Nonce Extension



