ICS 35.040
L 80

中华人民共和国国家标准

GB/T 25059—2010

信息安全技术　公钥基础设施 简易在线证书状态协议

Information security technology—Public Key Infrastructure—Simple Online Certificate Status Protocol

2010-09-02 发布　　2011-02-01 实施

中华人民共和国国家质量监督检验检疫总局
中国国家标准化管理委员会　发布

目　次

前　言

本标准的附录A为资料性附录，附录B为规范性附录。

本标准由全国信息安全标准化技术委员会(TC 260)提出并归口。

本标准主要起草单位：上海信息安全工程技术研究中心、国家信息安全工程技术研究中心。

本标准主要起草人：袁峰、郭晓雷、杨恒亮、谢安明、李增欣、苏瑞丹。

本标准责任专家：袁文恭。

引　言

在基于PKI的众多应用中，存在这种情况，某个应用系统的服务器在完成自身的功能时，需要进行大量实时的证书状态查询操作。在这种情况下，应用服务器对证书状态查询操作的性能与效率要求比较高，如果按照标准OCSP协议进行操作，协议数据单元复杂，签名和验签操作的开销都使得证书状态的查询操作成为应用服务器性能的瓶颈。而在实际应用的过程中，应用服务器往往和OCSP服务器位于同一可信网络或网段中，所以，为消除这一性能瓶颈，我们需要将标准OCSP协议进行简化，设计实现一个轻量级的证书状态查询协议。

信息安全技术　公钥基础设施
简易在线证书状态协议

1　范围

本标准规定了一种简易的在线证书状态协议——SOCSP。该协议可作为标准 OCSP 协议的补充。本标准主要描述了以下内容：

a）　具体描述了简易在线证书状态协议的请求形式；

b）　具体描述了简易在线证书状态协议的响应形式；

c）　分析了处理简易在线证书状态协议响应时可能出现的各种异常情况；

d）　说明了简易在线证书状态协议基于超文本传输协议(HTTP)的应用方式。

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。

2　规范性引用文件

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

GB/T 19713—2005　信息技术　安全技术　公钥基础设施　在线证书状态协议

3　术语和定义

下列术语和定义适用于本标准。

3.1

证书序列号　certificate serial number

由证书认证机构产生的唯一对应于每个证书的编号。

3.2

哈希　hash

将值从一个大的(可能很大)定义域映射到一个较小值域的(数学)算法。

3.3

请求者　requester

申请在线证书状态查询服务的主体。

3.4

响应者　responder

提供在线证书状态查询服务的主体。

4　缩略语

下列缩略语适用于本标准。

CRL　证书撤销列表(Certificate Revocation List)

DER　特异编码规则(Distinguished Encoding Rules)

HTTP　超文本传输协议(Hypertext Transfer Protocol)

MAC　信息验证码(Message Authentication Codes)

HMAC 带加密的散列消息验证码(Hash Message Authentication Codes)
LDAP 轻量目录访问协议(Lightweight Directory Access Protocol)
PKI 公开密钥基础设施(Public Key Infrastructure)
OCSP 在线证书状态协议(Online Certificate Status Protocol)
OID 对象标识(Object Identifier)
SMTP 简单邮件发送协议(Simple Mail Transfer Protocol)
SSL 安全套接口层协议(Secure Socket Layer protocol)
SOCSP 简易在线证书状态协议(Simple Online Certificate Status Protocol)
TLS 传输层安全协议(Transport Layer Security protocol)

5 总则

5.1 概述

本标准规定了检查证书状态的应用程序和提供证书状态查询的服务器之间使用简易在线证书状态协议需要交换的数据。应用环境要求 SOCSP 拥有独立的信息数据源。

5.2 请求

SOCSP 请求包含以下数据:

a) 协议版本;
b) 服务请求;
c) 目标证书标识符。

对某个请求的回应,SOCSP 响应者应确定:

a) 报文格式是否正确;
b) 响应者是否配置了所要求的服务;
c) 请求是否包含响应者需要的信息。

如果上述任何一个条件未满足,则 OCSP 响应者将发出一个出错信息;否则,将返回一个明确的响应。

5.3 响应

与标准 OCSP 协议类似,SOCSP 响应可以有不同类型。SOCSP 响应由响应类型和响应实体的字节组成。响应器对所有明确的响应报文都应进行 MAC 计算。

明确的响应消息由如下内容组成:

a) 响应语法的版本;
b) 对请求中每个证书的响应;
c) MAC 算法的 OID;
d) 响应的 MAC。

对请求中每个证书的响应由如下内容组成:

a) 目标证书标识符;
b) 证书状态值;
c) 本标准对证书状态值规定了如下明确的响应标识符:
   1) Good(好):表示对状态查询的肯定响应。如果客户端没有使用时间戳服务器和有限期验证的安全策略,肯定的响应只能说明证书未被撤销,但并不能说明证书已被发布或产生响应的时间是在证书有效性时间范围内。响应扩展可用于传输响应者作出的关于证书状态信息的附加声明,例如发布的肯定声明、有效性等。
   2) Revoked(已撤销):表示证书已被(永久地或临时地)撤销。
   3) unknown(不明):表示无此证书记录。

5.4 异常情况

当出现出错时，SOCSP 响应者返回某个出错消息。出错消息可以是下列几类：

a) malformedRequest(不完整的申请)：SOCSP 响应者(服务器)接收到的请求没有遵循 SOCSP 语法；

b) internalError(内部错误)：SOCSP 响应者处于非协调的工作状态，应当向另一个响应者再次进行询问；

c) unauthorized(未授权)：该查询是由未授权请求者向响应者提出的。

6 功能要求

6.1 协议设计目标

SOCSP 作为 GB/T 19713—2005 在线证书状态协议的补充，有它所适合的特定应用环境。在设计与实现过程中，主要考虑以下目标：

a) 密码运算开销小(消除标准 OCSP 协议大的开销)——去除标准 OCSP 中的非对称密码算法运算，代之以 MAC 计算；

b) 协议数据单元简易，易于处理——去除复杂的协议数据单元定义，尤其是扩展项；

c) 基于 HTTP 协议——具备基于 HTTP 协议传输的能力。与标准的 OCSP OVER HTTP 工作方式类似，宜采用 POST 方法。

通过扩展 MIME 注册完成与标准 OCSP 协议实现集成。响应器在接收到基于 HTTP 协议的 OCSP 请求包时可以判断它属于那种类型的协议：标准的 OCSP 协议或简易版本的 OCSP 协议，从而进行相应的处理。

6.2 MIME 注册

标准 OCSP 协议的请求和响应的 MIME 注册分别是 application/ocsp-request 和 application/ocsp-response。与其相对应，定义 SOCSP 协议的请求和响应的 MIME 注册为 application/ocsp-thin-request 和 application/ocsp-thin-response。根据所使用的传输机制(例如：HTTP、SMTP、LDAP 等)，实际的消息格式可能会发生相应的变化，参见附录 A。

7 具体协议

7.1 约定

本标准采用抽象语法记法(ASN.1)来描述具体协议内容，见附录 B。对于签名计算来说，要签名的数据依据 ASN.1 抽象语法记法，采用 DER 高级编码规则实现的。

如果无特殊说明，默认使用 ASN.1 显式记法。

7.2 请求

本条规定了符合 ASN.1 的请求语法。请求语法如下：

```
OCSPThinRequest     ::=     SEQUENCE {                          ——请求数据
    tbsThinRequest          TBSThinRequest,                     ——请求信息
    mac                     MacData                             ——请求信息的 MAC 值
}

TBSThinRequest      ::= SEQUENCE {
    version                 [0] EXPLICIT Version DEFAULT v1,    ——版本号
    random                  BIT STRING,                         ——随机数
    serialList              SEQUENCE OF CertificateSerialNumber ——查询证书的序列号
}
```

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

MacData ::= SEQUENCE {
    mac                     DigestInfo,
    macSalt                 OCTET STRING,
    iterations              INTEGER DEFAULT 1    ——默认为 1,hash 的反复次数
}

DigestInfo ::= SEQUENCE {
digestAlgorithm             DigestAlgorithmIdentifier, ——hash 算法
digest                      Digest
}
Digest ::= OCTET STRING
```

MAC 计算是基于 TBSThinRequest 的结构的 DER 编码,用共享密钥与 macSalt 的模二加作为 MAC 的初始向量,macSalt 保证相同数据 MAC 的结果不相同。MAC 算法由 digestAlgorithm 标识。random 的长度≥128 比特。支持符合国家规定的 HMAC 算法。

7.3 响应

本条规定了符合 ASN.1 的响应语法。tbsThinResponse 在本标准中定义为待进行 MAC 计算的数据。响应语法如下:

```
OCSPThinResponse   ::= SEQUENCE {
    tbsThinResponse     TBSThinResponse,                    ——响应信息
    mac                 MacData                             ——响应信息的 MAC 值
}

TBSThinResponse     ::=SEQUENCE {
    version             [0] EXPLICIT Version DEFAULT v1,
    random              BIT STRING,
    thinResponseStatus  [0] EXPLICIT OCSPThinResponseStatus,
    resultList          SEQUENCE OF ResultList OPTIONAL ——响应列表。非成功查询
                                                        时 resultList 项无值
}

Version             ::= INTEGER {     v1(0)     }

OCSPThinResponseStatus     ::= ENUMERATED {
    successful          (0),                                ——响应被有效确认
    malformedRequest    (1),                                ——不完整的请求
    internalError       (2),                                ——内部错误
    unauthorized        (3)                                 ——未授权
}
ResultList ::= SEQUENCE {
```

```
        serial              CertificateSerialNumber,        ——证书序列号
        certStatus          CertStatus,                     ——证书状态
        certStatusTime      GeneralizedTime                 ——证书状态时间

}
CertStatus ::= CHOICE {
        good                [0] IMPLICIT NULL,              ——未被撤销
        revoked             [1] IMPLICIT RevokedInfo,       ——已被撤销
        unknown             [2] IMPLICIT UnknownInfo        ——不明}

MacData ::= SEQUENCE {
        mac                 DigestInfo,
        macSalt             OCTET STRING,
        iterations          INTEGER DEFAULT 1               ——默认为 1,hash 的反复次数
}

DigestInfo ::= SEQUENCE {
        digestAlgorithm     DigestAlgorithmIdentifier,
        digest              Digest
}

Digest ::= OCTET STRING
```

MAC 的计算条件与请求包的计算条件一致，响应中的 random 与请求中的 random 相同。一个 SOCSP 响应至少由一个指明先前请求的处理状态的 thinResponseStatus 字段构成。如果 thinResponseStatus 的值是某个出错条件，则不设置 responseContent。

MAC 计算是基于 Response 结构的 DER 编码进行的。

7.4 MAC 算法

SOCSP 中 MAC 算法由 digestAlgorithm 标识。应支持符合国家规定的 HMAC 算法。

本标准中涉及的客户端和服务端使用的密码算法应采用国家主管部门批准的相关算法。

8 安全考虑

由于 SOCSP 简化了 OCSP 中复杂的签名和验证，应特别关注 SOCSP 的安全使用。SOCSP 作为标准 OCSP 协议的补充，在实际应用当中，如何对检查证书状态的应用程序和提供证书状态查询的服务器之间所共享的密钥进行保护很重要。本标准推荐，在实际应用中 HMAC 的有关参数应根据使用情况适时进行变化。

附 录 A
（资料性附录）
HTTP 上的 SOCSP

A.1 请求

基于 HTTP 的 SOCSP 请求可以使用 GET 或 POST 方法来提交。为了使得 HTTP 缓存生效，较小的请求(经过编码后小于 255 字节)可以用 GET 方法来提交。如果 HTTP 缓存不是很重要，或者请求大于或等于 255 字节，那么请求应该用 POST 方法来提交。在保密性是一个重要需求的时候，基于 HTTP 的 SOCSP 会话可以用 TLS/SSL 或其他底层的协议来保障其安全性。

一个使用 GET 方法的 OCSP 请求按如下方式进行构造：

GET {url}/{url-encoding of base-64 encoding of the DER encoding of the OCSPThinRequest}

其中 {url}可以从 AuthorityInfoAccess 的值或者 OCSP 客户端的本地配置获得。

一个使用 POST 方法的 OCSP 请求按如下方式进行构造：

Content-Type 头具有值“application/ocsp-thin-request”，而消息体是 OCSPThinRequest 的 DER 编码的二进制值。

注：实施者只使用 POST 方法，使用 POST 方法还可避免 HTTP 缓存机制带来的麻烦。

A.2 响应

一个基于 HTTP 的 SOCSP 响应由以下方式定义：

Content-Type 头具有值“application/ocsp-thin-response”，Content-Length 头应该指定响应的长度，而消息体是 OCSPThinResponse 的 DER 编码的二进制值。其他的不能被客户端识别的 HTTP 头可能存在在响应中，可以被忽略。

附 录 B
（规范性附录）
采用 ASN.1 定义的 SOCSP

```
SOCSP DEFINITIONS EXPLICIT TAGS::=
BEGIN
IMPORTS
        -- Directory Authentication Framework (X.509)
                Certificate,AlgorithmIdentifier,CRLReason
                FROM AuthenticationFramework { joint-iso-itu-t ds(5)
                        module(1) authenticationFramework(7) 3 }
        -- PKIX Certificate Extensions
                AuthorityInfoAccessSyntax
                FROM PKIX1Implicit88 {iso(1) identified-organization(3)
                        dod(6) internet(1) security(5) mechanisms(5) pkix(7)
                        id-mod(0) id-pkix1-implicit-88(2)}
                Name,GeneralName,CertificateSerialNumber,Extensions,
                id-kp,id-ad-ocsp
                FROM PKIX1Explicit88 {iso(1) identified-organization(3)
                        dod(6) internet(1) security(5) mechanisms(5) pkix(7)
                        id-mod(0) id-pkix1-explicit-88(1)}
        -- Cryptographic Message Syntax (CMS)
                IssuerAndSerialNumber
                FROM { iso(1) member-body(2) us(840) rsadsi(113549)
                        pkcs(1) pkcs-9(9) smime(16) modules(0) cms-2001(14)}

OCSPThinRequest         ::=     SEQUENCE {
    tbsThinRequest              TBSThinRequest,
    Mac                         MacData
}

TBSThinRequest          ::=     SEQUENCE {
    version                     [0] EXPLICIT Version DEFAULT v1,
    random                      BIT STRING
    serialList                  SEQUENCE OF CertificateSerialNumber
}

Version                 ::=     INTEGER {    v1(0)    }

MacData   ::= SEQUENCE {
    mac                         DigestInfo,
    macSalt                     OCTET STRING,
```

```
        iterations                  INTEGER DEFAULT 1
    }

    DigestInfo  ::= SEQUENCE {
    digestAlgorithm                 DigestAlgorithmIdentifier,
    digest                          Digest
    }

    Digest  ::= OCTET STRING

    OCSPThinResponse    ::=         SEQUENCE {
        tbsThinResponse             TBSThinResponse,
        Mac                         MacData
    }

    TBSThinResponse     ::=         SEQUENCE {
        version                     [0] EXPLICIT Version DEFAULT v1,
        random                      BIT STRING,
        thinResponseStatus          [0] EXPLICIT OCSPThinResponseStatus,
        resultList                  SEQUENCE OF ResultList OPTIONAL
    }

    Version             ::=         INTEGER {   v1(0)   }

    OCSPThinResponseStatus  ::=     ENUMERATED {
        successful                  (0),
        malformedRequest            (1),
        internalError               (2),
        unauthorized                (3)
    }
    ResultList ::= SEQUENCE {
        serial                      CertificateSerialNumber,
        certStatus                  CertStatus,
        certStatusTime              GeneralizedTime

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

    MacData ::= SEQUENCE {
```

```
    mac                     DigestInfo,
    macSalt                 OCTET STRING,
    iterations              INTEGER DEFAULT 1
}

DigestInfo ::= SEQUENCE {
digestAlgorithm             DigestAlgorithmIdentifier,
digest                      Digest
}

Digest ::= OCTET STRING
```

中华人民共和国
国家标准
信息安全技术 公钥基础设施
简易在线证书状态协议
GB/T 25059—2010
*
中国标准出版社出版发行
北京复兴门外三里河北街16号
邮政编码:100045
网址 www.spc.net.cn
电话:68523946 68517548
中国标准出版社秦皇岛印刷厂印刷
各地新华书店经销
*
开本 880×1230 1/16 印张 1 字数 21 千字
2010年11月第一版 2010年11月第一次印刷
*
书号:155066·1-40463 定价 18.00 元

GB/T 25059-2010

打印日期:2010年12月9日 F009