PKIX Working Group
draft-malpani-rcsp-00.txt
Expires in 6 months

Ambarish Malpani(ValiCert)
Carlisle Adams (Entrust Technologies)
Rich Ankney (CertCo)
Slava Galperin (Netscape)

March, 1998

Internet Public Key Infrastructure
Real Time Certificate Status Protocol - RCSP
<draft-malpani-rcsp-00.txt>

Status of this Memo

1. Abstract

The protocol conventions described in this document satisfy some of the operational requirements of the Internet Public Key Infrastructure (PKI). This document specifies a protocol useful in determining the current status of a digital certificate without the use of CRLs. Additional mechanisms addressing PKIX operational requirements are specified in separate documents.

Section 2 provides an overview of the protocol. Section 3 goes through the functional requirements, while section 4 provides the details of the protocol. In section 5 we cover security issues with the protocol. Appendix A demonstrates RCSP over HTTP.

Please send comments on this document to the ietf-pkix@tandem.com mail list.

2. Protocol Overview

In lieu of or as a supplement to checking against a periodic CRL, it MAY be necessary to obtain timely status regarding a certificate's

revocation status (cf. PKIX Part 1, Section 3.3). Examples include high-value fund transfers or the compromise of a highly sensitive key.

The Real Time Certificate Status Protocol (RCSP) enables applications to determine the revocation state of an identified certificate. RCSP may be used to satisfy some of the operational requirements of providing more timely revocation information than is possible with CRLs. An RCSP client issues a status request to an RCSP responder and suspends acceptance of the certificate in question until the responder provides a response.

This protocol specifies the data that needs to be exchanged between an application checking the revocation status of a certificate and the server providing that status.

2.1 Request

An RCSP request contains the following data:

- protocol version
- service request
- target identifier for one or more certificates to be checked
- optional extensions which MAY be processed by the RCSP responder

Upon receipt of a request, an RCSP responder determines if: 1) the message is well formed, 2) the responder is configured to provide the requested service, and 3) the responder can perform the requested service for the certificate in question. If any of the prior conditions are not met, the RCSP responder produces an error message; otherwise, it returns a definitive response.

2.2 Response

All definitive responses SHALL be digitally signed. The key used to sign definitive responses need not be the same key used to sign the certificate. The key used to sign the response MUST belong to one of the following:

- a Trusted Responder, whose public key is already known to the requestor
- the CA who issued the certificate in question
- an Authorized Responder, with a certificate which has a special extension
authorizing it to sign RCSP responses

A definitive response message is composed of:

- response type identifier (to allow for different response types)
- version of the response
- name of the responder
- responses for each of the certificates in a request
- optional extensions

- signature algorithm OID
- signature computed across the hash of the response

The response for each of the certificates in a request consists of

- target certificate identifier
- certificate status value
- response validity interval
- optional extensions

This specification defines the following definitive response indicators for use in the certificate status value:

- notRevoked
- revoked
- onHold

The notRevoked state indicates that the certificate is not revoked. It does not necessarily mean that the certificate was ever issued. Nor does it mean that the certificate is in its validity interval. A notRevoked state by an RCSP responder DOES NOT absolve the application of the responsibility of checking that the certificate is in its validity period and has been correctly signed.

The revoked state indicates that the certificate has been revoked.

The onHold state corresponds to valid certificates that are operationally suspended in accordance with PKIX Part 1.

In case of errors, the RCSP Responder may return an error message. Errors
can be of the following types:

- malformedRequest
- requestorUnauthorized
- internalError
- tryLater

A server produces the malformedRequest response if the request received does not conform to the RCSP syntax.

The response requestorUnauthorized is used in cases where the server does not consider the client authorized to query it. Authorization data is not explicitly a part of the request or this protocol. However, certain responders may choose to require clients to ship an authorization token with the request and may choose to refuse service to clients that do not ship a correct authorization token.

The response internalError indicates that the RCSP responder reached an inconsistent internal state. The query should be retried, potentially with another responder.

The response tryLater is produced in any circumstance in which the server has received a well formed RCSP request but is unable to process it.

2.3 Response Pre-production

The response validity interval noted in the prior section is composed of a {producedAt, nextUpdate} pair of elements in the response syntax. Section 4.2 provides details of the response syntax.

RCSP responders MAY pre-produce signed responses specifying the current status of certificates at the time the response was produced. The time at which the response was known to be correct SHALL be specified in the producedAt field of the response. This time is not necessarily the same as the time at which the response was produced - e.g. if the responder obtains a CRL from a CA and creates pre-produced responses, the producedAt time should specify the thisUpdate time in the CRL.

The producer of the response MAY include a value for nextUpdate. The exact interval between producedAt and nextUpdate for a given response is a matter of local security and operational policy. If the nextUpdate field is not present, the response is known to be correct at the producedAt time. No assertions are being made about the current state of the certificate, nor are any recommendations being made as to when the requestor should check again with the responder. If the value of nextUpdate is set, it is just a hint, not a guarantee, of when the responder expects to have new information about that certificate's status.

If responses are pre-produced, then for a given certificate, the periodicity of this pre-production SHOULD match the response validity interval of the most recently produced response.

2.4 Delegation of the task of RCSP responses to an Authorized Responder

One or more CAs may decide to delegate the task of producing RCSP response to a third party (an Authorized Responder). In that case, each of those CAs should provide the Authorized Responder with a certificate
that includes a special extension authorizing the holder to issue RCSP responses.

3. Functional Requirements

3.1 Certificate Content

In order to convey to RCSP clients a well-known point of information access, CAs SHALL provide the capability to include the AuthorityInfoAccess extension (defined in PKIX Part 1, section 4.2.2.1) in certificates that can be checked using RCSP.

CAs that support an RCSP service, either hosted locally or provided by an Authorized Responder, MAY provide a value for a uniformResourceIndicator (URI) accessLocation and the OID value id-ad-rcsp for the accessMethod in the AccessDescription SEQUENCE. The value of the accessLocation field in the subject certificate corresponds to the URL placed into an RCSP request. Alternatively, the accessLocation for the RCSP provider may be configured locally at the RCSP client (e.g., in cases where the RCSP provider is a trusted party for the particular client, whose job is to aggregate revocation information from all trusted CAs).

```
id-ad-rcsp OBJECT IDENTIFIER ::= {id-ad ?}
```

3.2 Error Responses

Upon receipt of a request which fails to parse, the receiving RCSP responder SHALL respond with an error message. Error responses MAY be signed.

3.3 Signed Response Acceptance Requirements

Prior to accepting a signed response as valid, RCSP clients SHALL confirm that:

- The certificate identified in a received response corresponds to that which was identified in the corresponding request.
- The signature on the response is valid.
- The identity of the signer matches the intended recipient of the request.
- The signer is currently authorized to sign the response.

4. Detailed Protocol

The ASN.1 syntax imports terms defined in the X.509 Certificate and CRL Profile Internet Draft. For signature calculation, the data to be signed is
encoded using the ASN.1 distinguished encoding rules (DER) [X.690].

ASN.1 EXPLICIT tagging is used as a default unless specified otherwise.

The terms imported from elsewhere are:
Version, Extensions, CertificateSerialNumber, SubjectPublicKeyInfo, Name,
AlgorithmIdentifier, GeneralizedTime

4.1 Request Syntax

```
RCSPRequest     ::=     SEQUENCE {
        version                 [0]     EXPLICIT Version DEFAULT v1,
        hashAlgorithm                   AlgorithmIdentifier,
```

```
        requestList                     SEQUENCE OF Request,
        requestExtensions       [1]     EXPLICIT Extensions OPTIONAL
}

Request                 ::=     CHOICE {
        certID                  [0]     EXPLICIT CertID,
        cert                    [1]     EXPLICIT Certificate
}

CertID                  ::=     SEQUENCE {
        issuerNameAndKeyHash            Hash,
        serialNumber                    CertificateSerialNumber,
}

IssuerNameAndKey                ::=     SEQUENCE {
        issuer                          Name,
        issuerPublicKey                 SubjectPublicKeyInfo
}

Hash                    ::=     OCTET STRING --hash of IssuerNameAndKey--
```

4.2 Notes on the RCSP Request Syntax

The issuerNameAndKeyHash is computed by hashing an IssuerNameAndKey field constructed for the CA in question using a cryptographic hash function (e.g., SHA-1) specified as the hashAlgorithm in the request. The primary reason to use both the name and the public key to identify the issuer is that it is possible that two CAs may choose to use the same Name (uniqueness in the Name is a recommendation that cannot be enforced). Two CAs will never, however, have the same public key unless the CAs either explicitly decided to share their private key, or the key of one of the CAs was compromised.

While it is possible to identify a certificate by sending over either the entire certificate or just a CertID, it is recommended that clients use just the CertID to reduce the size of both the request and the response. However, certain RCSP responders MAY require the entire certificate whose status is to be determined.

Support for extensions is OPTIONAL. The critical flag SHOULD NOT be set for any of them. This standard suggests several useful extensions in Section 4.5. Additional extensions MAY be defined in additional RFCs. Unrecognized extensions SHOULD be ignored.

4.2 Response Syntax

This section specifies the ASN.1 specification for a confirmation response. The actual formatting of the message could vary depending on the transport mechanism used (http, smtp, ldap, etc.).

4.2.1 ASN.1 Specification of the RCSP Response

```
RCSPResponse            ::=     SEQUENCE {
        responseStatus                  RCSPResponseStatus,
        responseBytes           [0]     EXPLICIT ResponseBytes OPTIONAL
}

RCSPResponseStatus      ::=     ENUMERATED {
        successful              ( 0 ),  --Response has valid confirmations--
        malformedRequest        ( 1 ),  --Illegal confirmation request--
        requestorUnauthorized   ( 2 ),  --User not authorized to use issuer--
        internalError           ( 3 ),  --Internal error in issuer--
        tryLater                ( 4 )   --Try again later--
}

ResponseBytes           ::=     SEQUENCE {
        responseType                    OBJECT IDENTIFIER,
        response                        OCTET STRING
}
```

If the responseStatus is one of the error conditions, responseBytes are not set.

For a basic RCSP responder, responseType will be id-pkix-rcsp-basic, where:
id-pkix-rcsp OBJECT IDENTIFIER ::= { id-pkix ? }
id-pkix-rcsp-basic OBJECT IDENTIFIER ::= { id-pkix-rcsp ? }

The response will be the DER encoding of BasicRCSPResponse

```
BasicRCSPResponse       ::=     SEQUENCE {
        tbsResponseData                 ResponseData,
        signatureAlgorithm              AlgorithmIdentifier,
        signature                       BIT STRING,
        certs                   [1]     EXPLICIT SEQUENCE OF Certificate
                                                OPTIONAL
}

ResponseData            ::=     SEQUENCE {
        version                 [0]     EXPLICIT Version DEFAULT v1,
        responderName                   Name,
        responses                       SEQUENCE OF SingleResponse,
        responseExtensions      [1]     EXPLICIT Extensions OPTIONAL
}

SingleResponse          ::=     SEQUENCE {
        request                         Request,
        certStatus                      CertStatus,
        producedAt                      GeneralizedTime,
        nextUpdate              [0]     EXPLICIT GeneralizedTime OPTIONAL,
        singleExtensions        [2]     EXPLICIT Extensions OPTIONAL
}
```

```
CertStatus              ::=     CHOICE {
        certStatusType          [0]    EXPLICIT CertStatusType
                                        (notRevoked | onHold),
        statusWithTime          [1]    EXPLICIT StatusWithTime
}

StatusWithTime          ::=     SEQUENCE {
        certStatusType                  CertStatusType (revoked),
        time                            GeneralizedTime
}


CertStatusType          ::=     ENUMERATED {
        notRevoked              (0),    --This serial number is not revoked--
        revoked                 (1),    --Serial number was revoked--
        onHold                  (2)     --Cert is on hold--
}
```

4.2.2 Notes on RCSP Responses

If the certStatusType is revoked, time is the time of revocation of the certificate.

The producedAt and nextUpdate fields define a recommended validity interval. This interval corresponds to the {thisUpdate, nextUpdate} interval in CRLs. Responses whose nextUpdate value is earlier than the local system time value SHOULD be considered unreliable. Responses whose producedAt time is earlier than the local system time SHOULD be considered unreliable. Responses where the nextUpdate value is not set are equivalant to a CRL with no time for nextUpdate (see section 2.3).

The signature should be computed on the hash of the DER encoding of the ResponseData.

4.3 Authorized Responders

One or more CAs may designate an Authorized RCSP Responder, by issuing a special certificate. The Authorized Responder will have the right to issue RCSP responses on behalf of that CA as long as that certificate does not expire. The certificates for Authorized Responders SHALL contain a non-critical extension, proving the authorization. The extension identifier will be id-kp-rcsp, while the value will be the DER encoding of the integer 1. The certificate of an Authorized RCSP Responder cannot be revoked. It is recommended that the life of this certificate be kept short and every effort be made to protect its private key atleast as carefully as that of the CA.

```
id-kp-rcsp OBJECT IDENTIFIER ::= {id-kp ?}
```

4.4 Mandatory and Optional Cryptographic Algorithms

Clients that request RCSP services SHALL be capable of processing responses signed used DSA keys identified by the DSA sig-alg-oid specified in section 7.2.2 of PKIX Part 1. Clients SHOULD also be capable of processing RSA signatures as specified in section 7.2.1 of PKIX Part 1. RCSP responders SHALL support the SHA1 hash algorithm.

4.5 Extensions

This section defines the way to support some commonly requested tasks. Support for all extensions is OPTIONAL, so critical SHOULD NOT be set for any of these extensions.

4.5.1 Nonce

The nonce cryptographically binds a request and a response to prevent replay attacks. The nonce is included as one of the requestExtensions in requests, while in responses it would be included as one of the responseExtensions. In both the request and the response, the nonce will be identified by the object identifier id-pkix-rcsp-nonce, while the extnValue is the value of the nonce.

```
id-pkix-rcsp-nonce OBJECT IDENTIFIER ::= { id-pkix-rcsp ? }
```

4.5.2 Signed Requests

This extension allows the requestor to sign a request. The requestor includes an extension that has the signatureIdentifier, the actual bits of the signature and a sequence of certificates to allow the RCSP responder to verify the signature. The data to be signed is just the basic request (none of the extensions). The RCSP Responder can verify the signature, potentially using certificates that have been included with the extension. The signature on a request will be identified by id-pkix-rcsp-signature, while the value will be SignatureData, where:

```
id-pkix-rcsp-signature OBJECT IDENTIFIER ::= { id-pkix-rcsp ? }
SignatureData           ::=     SEQUENCE {
        signatureAlgorithm      AlgorithmIdentifier,
        signature               BIT STRING,
        certs           [0]     EXPLICIT SEQUENCE OF Certificate OPTIONAL
}
```

4.5.3 CRL References

It MAY be desirable for the RCSP responder to indicate the CRL on which a revoked or onHold certificate is found. This can be useful where RCSP is used between repositories, and also as an auditing mechanism. The CRL may be specified by a URL (the URL at which the CRL is available), a number (CRL number) or a time (the time at which the relevant CRL was created). These extensions will be specified as singleExtensions.

The identifier for this extension will be id-pkix-rcsp-crl, while the value
will be CrlID.

```
id-pkix-rcsp-crl OBJECT IDENTIFIER ::= { id-pkix-rcsp ? }
CrlID           ::=     SEQUENCE {
        crlUrl          [0]     EXPLICIT IA5String OPTIONAL,
        crlNum          [1]     EXPLICIT INTEGER OPTIONAL,
        crlTime         [2]     EXPLICIT GeneralizedTime OPTIONAL
}
```

For the choice crlUrl, the IA5String will specify the URL at which the CRL is available. For crlNum, the INTEGER will specify the value of the CRL
number extension of the relevant CRL. For crlTime, the GeneralizedTime will indicate the time at which the relevant CRL was issued.

4.5.4 Acceptable Response Types

An RCSP client MAY wish to specify the kinds of response types it understands. To do so, it SHOULD use an extension with the OID id-pkix-rcsp-response, and the value AcceptableResponses. The id's included in AcceptableResponses are the OIDs of the various response types this client can accept (e.g., id-pkix-rcsp-basic).

```
id-pkix-rcsp-response OBJECT IDENTIFIER ::= {id-pkix-rcsp ?}
AcceptableResponses     ::=     SEQUENCE OF {
        id                              OBJECT IDENTIFIER
}
```

4.5.4 Other Extensions

CRL Entry Extensions - specified in Section 5.3 of PKIX part I - are also
supported as singleExtensions.

5. Security Considerations

For this service to be effective, systems using certificates must connect to the certificate status service provider. In the event such a connection cannot be obtained, these systems could implement CRL processing logic as a fall-back position.

If a CA authorizes another public key to sign RCSP responses, compromise of that key is as serious as the compromise of the CA's key as long as the authorization is valid. An authorized RCSP responder's key cannot be revoked.

A denial of service vulnerability is evident with respect to a flood of queries constructed to produce error responses. The production of a cryptographic signature significantly affects response generation cycle time, thereby exacerbating the situation. Unsigned error

responses can be produced more rapidly and thus reduce the danger of this
attack. However, unsigned error responses open up the protocol to another denial of service attack, where the attacker sends false error responses.

The use of precomputed responses allows replay attacks in which an old (notRevoked) response is replayed prior to its expiration but after the certificate has been revoked. To reduce this vulnerability, it is recommended that the period between producedAt and nextUpdate be kept as small as possible.

6. Acknowledgements

This protocol uses many ideas from the Online Certificate Status Protocol (OCSP), developed by Mike Meyers (VeriSign) and Rich Ankney (CertCo). We have also used comments from Marc Branchaud (XCert), Robert Zuccherato (Entrust) and Anil Gangolli (Structured Arts).

7. References

[HTTP] Hypertext Transfer Protocol -- HTTP/1.0. T. Berners-Lee, R. Fielding & H. Frystyk, RFC 1945, May 1996.

[MUSTSHOULD] Key words for use in RFCs to Indicate Requirement Levels, S. Bradner, RFC 2119, March 1997.

[URL] Uniform Resource Locators (URL), T. Berners-Lee, L. Masinter, M. McCahill, RFC 1738, December 1994.

8. Author's Address

Ambarish Malpani
ValiCert, Inc.
3160 W. Bayshore Drive
Palo Alto, CA 94303
ambarish@valicert.com

Carlisle Adams
Entrust Technologies
750 Heron Road, Suite E08
Ottawa, Ontario
K1V 1A7
Canada
cadams@entrust.com

Rich Ankney
CertCo, LLC
13506 King Charles Dr.
Chantilly, VA 20151
rankney@erols.com

Slava Galperin
Netscape Communications Corp.
MV-068
501 E. Middlefield Rd.
Mountain View, CA 94043
galperin@netscape.com

Appendix A

A.1 RCSP over HTTP

This section describes the formatting that will be done to the request and
response to support HTTP.

A.1.1 Request

An RCSP request is an HTTP 1.0 POST method. The Content-Type header has the value "application/rcsp-request" while the body of the message is the DER encoding of the RCSPRequest.

A.1.2 Response

An HTTP-based RCSP response is composed of the appropriate HTTP headers, followed by the DER encoding of the RCSPResponse. The Content-Type header has the value "application/rcsp-response". The Content-Length header SHOULD specify the length of the response. Other HTTP headers MAY be present and MAY be ignored if not understood by the requestor.