Internet Draft Michael
Myers
draft-ietf-pkix-ocspv2-02.txt TraceRoute
Security
March, 2001 Rich
Ankney
Expires in six months
Certco
Carlisle
Adams

Entrust
Stephen
Farrell

Baltimore
Carlin
Covey

Cylink

Online Certificate Status Protocol, version 2

draft-ietf-pkix-ocspv2-02.txt

Status of this memo

This document is an Internet-Draft and is in full conformance with all provisions of Section 10 of RFC 2026.

Internet-Drafts are working documents of the Internet Engineering Task Force (IETF), its areas, and its working groups. Note that other groups may also distribute working documents as Internet-Drafts.

Internet-Drafts are draft documents valid for a maximum of six months and may be updated, replaced, or obsoleted by other documents at any time. It is inappropriate to use Internet-Drafts as reference material or to cite them other than as "work in progress."

The list of current Internet-Drafts can be accessed at
http://www.ietf.org/ietf/1id-abstracts.txt

The list of Internet-Draft Shadow Directories can be accessed at
http://www.ietf.org/shadow.html.

1. Abstract

This document advances the specification of RFC 2560 [RFC2560] towards the acquisition of any data relevant to determining a digital certificate's reliability; interoperability with [RFC2560] is maintained. Such data include a certificate's revocation status, it's validation status and ancillary information supporting these assertions. Three services are defined:

Online Revocation Status (ORS); Delegated Path Validation (DPV); and Delegated Path Discovery (DPD). Ancillary extensions defined in [RFC2560] are also maintained. The means by which a certificate may be identified is also expanded.

The key words "MUST", "MUST NOT", "REQUIRED", "SHALL", "SHALL NOT", "SHOULD", "SHOULD NOT", "RECOMMENDED", "MAY", and "OPTIONAL" in this document (in uppercase, as shown) are to be interpreted as described in [RFC2119].

TABLE OF CONTENTS

2. Protocol Overview

OCSP is a online request-response PKI information acquisition protocol composed of generic envelope and a set of standardized request and response types. An OCSP request message is composed of protocol version number, a request type object identifier and other request data relevant to a particular request type. Requests may be digitally signed. Responses are correspondingly structured as an OID and associated response data. Responses may or may not be digitally signed depending on service request type or error processing.

A request-response pair defines a service type. The Online Revocation Status (ORS), Delegated Path Validation (DPV) and Delegated Path Discovery (DPD) service types are of particular interest.

The ORS service enables certificate processing software to obtain timely information regarding the revocation status of a certificate beyond that typically available through the use of CRLs. While useful to the deployment of this service in certain instances, CRLs are not mandatory to implement ORS.

The DPV service enables an environment to offload certificate path validation complexity to a centralized server, thereby yielding benefits to thin clients and enterprise security policy management. DPV provides client-side control points whereby a client may govern essential aspects of a server's technical behavior. These are:

a. OID-based specification of policies relevant to path validation logic;

b. specification of paths or partial paths to be respected by the server;

c. a means to specify a time context for historical references;

d. inclusion of revocation information, whether CRLs or OCSP; and

e. flags that stimulate a DPV server to return ancillary information, including:
the path used, the revocation information relied on, a timestamp on the entire process and an OID-based identification of the policy or policies enforced in the validation process.

DPV control points are optional elements. Minimally, the relationship between a
DPV client and a DPV server is an answer to the question "is this certificate valid now?" The control points enable a client to dynamically tailor its definition of "valid" in accordance with localized policies.

The DPD service enables certificate processing software to acquire the certificate and revocation data a DPV server might otherwise use, thereby yielding benefits in integrating a single PKI information access protocol across

a potentially diverse set of more general data retrieval mechanisms (e.g. X. 500,
LDAP, HTTP, FTP, etc.) As with DPV, the DPD service specification provides for optional client-side control points. These are:

a. an iterative mechanism that enables a client to walk through several technically valid paths to discover an path acceptable to the client's operational policies (should such multiple paths exist);

b. OID-based specification of policies relevant to path validation logic;

c. specification of paths or partial paths to be respected by the server;

d. a means to specify a time context for historical references; and

e. flags that stimulate a DPD server to return ancillary information, including:
a timestamp on the entire process; an OID-based identification of the policy or policies enforced in the path construction process; and whether or not the client prefers CRLs, OCSP responses or either (in some instances one or the other might not be available to a DPD server).

OCSP also provides for the use of extensions that enable specialization of service types towards unique circumstances. These extensions, originally defined by [RFC2560] and carried forward in this document, include:

a. Nonces useful for replay detection needs;

b. CRL references, enable clients to acquire the CRLs used;

c. Archive Cutoff specification applicable to historical references;

d. CRL entry extensions for a acquisition of specific CRL content; and

e. Service Locator mechanism that enables client-driven server redirection.

3. General Protocol Requirements

Service requests and responses SHALL conform to this Section 3.0. Service definitions MAY extend these requirements.

3.1 Requests

Requests SHALL include the following content:

```
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 OPTIONAL }

Version         ::=            INTEGER  {  v1(0), v2(1) }
```

A requestExtensions field SHALL ONLY be included if the requestExtensions field contains one or more values, else the field SHALL be omitted from OCSPRequest.

3.1.1 Service Identification

As a general rule, service request types are identified by an OID in the requestExtensions field of TBSRequest. An exception is made for the Online Revocation Status (ORS) service. Other than ORS, the requests of all OCSP-based
service SHALL be an OID identifying the requested service in the requestExtensions field of TBSRequest; he ORS service MAY be so identified. To determine if an request is or is not a request for ORS service, servers SHALL implement logic equivalent to the following:

a. Upon receipt of a request, examine the contents of the requestExtensions field.

b. If the requestExtensions field is absent, the request SHALL be considered an ORS service request.

c. If the requestExtensions field contains one or more OIDs, then if any one of those values matches the ORS service type OID defined in this document, the request SHALL be considered an ORS service request.

3.1.2 Certificate Identification

The ReqCert structure enables clients to identify a certificate in the means most suitable to the technical constraints of their local environment. This structure interoperably extends the CertID definition defined in [RFC2560]. The
following options are available:

```
ReqCert  ::= CHOICE {
   certID            CertID,
   issuerSerial      [0] IssuerandSerialNumber,
   pKCert            [1] Certificate,
   name              [2] GeneralName,
   certHash          [3] OCTET STRING}

CertID          ::=     SEQUENCE {
   hashAlgorithm       AlgorithmIdentifier,
   issuerNameHash      OCTET STRING, -- Hash of Issuer's DN
   issuerKeyHash       OCTET STRING, -- Hash of Issuers public key
   serialNumber        CertificateSerialNumber }
```

As noted, [RFC2560] interoperablity can be obtained through the use of the certID element of the ReqCert CHOICE. If certID is used in ReqCert, the value for version in the tbsRequest field of OCSPRequest SHALL be v1. If any other choice in ReqCert is used, the value for version SHALL be v2.

Clients claiming compliance to this specification SHALL support either issuerSerial OR PKCert. Servers claiming compliance to this specification SHALL
be capable of responding to both. Support for other options is STRONGLY ENCOURAGED.

The value for issuerNameHash SHALL be calculated over the DER encoding of the issuer's name field in the certificate being checked.

The value for issuerKeyHash SHALL be calculated over the value of the BIT STRING
subjectPublicKey key field (excluding tag and length) in the issuer's certificate.

The hash algorithm used for both these hashes, is identified in hashAlgorithm. serialNumber is the serial number of the certificate for which status is being requested.

3.1.3 Signed Requests

Service definitions MAY require the use of digital signatures over requests. When required, signed request SHALL be produced as follows:

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

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

Signed requests SHALL identify the requestor in the requestorName field of TBSRequest (see Section 3.1). Signed requests MAY include in the certs field of the Signature element certificates that assist the OCSP responder in verifying the requestor's signature.

3.2 Response Envelope

Responses SHALL contain a responseStatus field. A value for the responseBytes field SHALL be included if the value for responseStatus is "successful". A value for the responseBytes field MAY be included for responseStatus values other than "successful". The responseBytes field SHALL be composed of an OBJECT
IDENTIFIER identifying the response type and the bytes of the actual response encoded as an OCTET STRING.

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

OCSPResponseStatus ::= ENUMERATED {
   successful            (0),
   malformedRequest      (1),
```

```
   internalError         (2),
   tryLater              (3),
                       --(4) is not used for legacy reasons
   sigRequired           (5),
   unauthorized          (6),
   noMoreData            (7) }
```

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

3.2.1 Definition of Basic OCSP Response

This section defines syntax useful to both the ORS and DPV services. Other services MAY reuse this syntax as needs dictate. In cases where this syntax is used, service definitions SHALL provide a definition for the certStatus field of
SingleResponse.

```
BasicOCSPResponse       ::= SEQUENCE {
   tbsResponseData      ResponseData,
   signatureAlgorithm   AlgorithmIdentifier,
   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 OPTIONAL }

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

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

3.2.2 Error Responses

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

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

In the event that a server is operational but unable to return a status for the requested certificate, the "tryLater" response can be used to indicate that the service exists, but is temporarily unable to respond.

The response "sigRequired" is returned in cases where the server requires the client sign the request in order to construct a response.

The response "unauthorized" is returned in cases where the client is not authorized to make this query to this server.

The response "noMoreData" is returned in cases where the server has previously returned the last positive response to a related sequence of requests.

3.2.3 Signed Responses

Response messages containing only a value for responseStatus SHALL NOT be digitally signed. Response signatures SHALL be computed over the DER encoding of the tbsRequest structure.

3.2.4 Response Acceptance Criteria

Clients SHALL reject as invalid any response that does not satisfy all of the following criteria:

a. the certificate(s) identified in a received response matches that (or those) identified in the corresponding request;

b. the signature on the response is valid;

c. the identity of the signer matches the intended recipient of the request;

d. the signer is currently authorized to sign the response;

e. when available, the time at which the status being indicated is known to be correct is sufficiently recent; and

f. when available, the time at or before which newer information will be available is greater than the current time.

3.3 Certificate Content

Certificate-producing entities SHALL be capable of including the AuthorityInfoAccess extension (defined in [RFC2459], section 4.2.2.1) in certificates. Alternatively, the accessLocation for the OCSP provider may be configured locally at the OCSP client.

Certificate-producing entities supporting this protocol SHALL be capable of providing for the inclusion of a value for a uniformResourceIndicator (URI) accessLocation and the OID value id-ad-ocsp for the accessMethod in the AccessDescription SEQUENCE.

The value of the accessLocation field in the subject certificate defines the transport (e.g. HTTP) used to access the OCSP responder and may contain other transport dependent information (e.g. a URL).

3.4 Authorized Responders

The key that signs a certificate's status information need not be the same key that signed the certificate. It is necessary however to ensure that the entity

signing this information is authorized to do so. Therefore, a certificate's issuer SHALL either:

a. sign the OCSPresponses itself using a private key under the control of the issuer; or

b. explicitly delegate this authority to another entity. OCSP signing delegation SHALL be designated by the inclusion of id-kp-OCSPSigning in an extendedKeyUsage certificate extension included in the OCSP response signer's certificate. This certificate MUST be issued directly by the CA that issued the certificate in question.

Explicit delegation of OCSP signing authority SHALL be indicated by inclusion of the following value in the extendedKeyUsage extension of certificate needed to validate an OCSP response signature produced by such an entity:

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

Systems or applications that rely on OCSP responses SHALL be capable of detecting and enforcing use of the id-ad-ocspSigning value as described above. They MAY provide a means of locally configuring one or more OCSP signing authorities, and specifying the set of CAs for which each signing authority is trusted. They SHALL reject the response if the certificate required to validate the signature on the response fails to meet at least one of the following criteria:ee

a. Matches a local configuration of OCSP signing authority for the certificate in question;

b. is a certificate corresponding to a private key under control of the CA that issued the certificate in question; or

c. includes a value of id-ad-ocspSigning in an ExtendedKeyUsage extension and is issued by the CA that issued the certificate.

Additional acceptance or rejection criteria may apply to either the response itself or to the certificate used to validate the signature on the response.

Since an Authorized Responder provides status information for one or more CAs, OCSP clients need to know how to check that an Authorized Responder's certificate has not been revoked. CAs may choose to deal with this problem in one of three ways:

- A CA may specify that an OCSP client can trust an Authorized Responder for the lifetime of the responder's certificate. The CA does so by including the extension id-pkix-ocsp-nocheck. This SHOULD be a non-critical extension. The value of the extension is NULL. CAs issuing such a certificate should realize that a compromise of the Authorized Responder's key, is as serious as the compromise of a CA key

used to sign CRLs, at least for the validity period of this certificate. CAs may choose to issue this type of certificate with a very short lifetime and renew it frequently.

```
id-pkix-ocsp-nocheck OBJECT IDENTIFIER ::= { id-pkix-ocsp 5 }
```

- A CA may specify how the responder's certificate is to be checked for revocation. This can be done using CRL Distribution Points if the check should be done using CRLs or CRL Distribution Points, or Authority Information Access if the check should be done in some other way. Details for specifying either of these two mechanisms are available in [RFC2459].

- A CA may choose not to specify any method of revocation checking for the responder's certificate, in which case, it would be up to the OCSP client's local security policy to decide whether that certificate should be checked for revocation or not.

3.5 Use of ASN.1 Syntax

The ASN.1 syntax used in this document imports terms defined in [RFC2459] and [RFC2630]. 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: Extensions, CertificateSerialNumber, SubjectPublicKeyInfo, Name, AlgorithmIdentifier, CRLReason and IssuerAndSerialNumber.

3.6 Application to Attribute Certificates

OCSP MAY be used without modification to provide status on attribute certificates (ACs) conforming to [ACPROF]. Note that since [ACPROF], section 4.2.3, mandates that the issuer field uses a single X.500 name to identify the AC issuer, and also ([ACPROF], section 4.5) mandates that an AC issuer cannot also be a PKC issuer (at least using the same name/key), there are no syntactic changes required to support ACs. An OCSP responder that only supports PKCs will treat a request for status on an AC in the same way as it would treat a request for an issuer/serial combination for which it had no information. That is, a conforming OCSP responder need not have any specific ability to handle ACs.

3.7 Mandatory and Optional Cryptographic Algorithms

Clients and servers SHALL support the RSA signature algorithm in accordance with
RFC 2437 [RFC2437] and the SHA1 hashing algorithm as defined in FIPS Pub 180-1 [SHA1]. DSA signature support MAY be provided. If provided, this algorithm SHALL
be implemented in accordance with FIPS Pub 186 [DSS].

4. Service Deployment Requirements

This document defines three standard services: Online Revocation Status (ORS),

Delegated Path Validation (DPD) and Delegated Path Discovery (DPD). Other services MAY be defined for environments with more local needs. Environments MAY deploy any or all of these services in any combination. That is, a server MAY be implemented and operated to only perform the DPD service.Equivalently, delivery of the ORS service is NOT required to deliver the DPV service. Servers
SHOULD be capable of delivering all three services.

5. Online Revocation Status

This section defines the Online Revocation Status (ORS) service. This service enables certificate processing software to determine if a certificate is or is not revoked. This information MAY be recorded in non-CRL formats. While useful to the deployment of this service in certain instances, CRLs are NOT required to implement ORS.

Responders SHALL be capable of producing responses of the id-pkix-ocsp-basic response type. Correspondingly, OCSP clients SHALL be capable of receiving and processing responses of the id-pkix-ocsp-basic response type.

The value for response SHALL be the DER encoding of BasicOCSPResponse.

5.1 Request

An ORS request SHALL conform to the specifications detailed in Section 3.1 of this document. An ORS request MAY be digitally signed. Clients are STRONGLY ENCOURAGED to include a value of id-pkix-ocsp-ors-req as a value for the requestExtensions field. Servers claiming compliance to this OCSPv2 specification SHALL: 1) be capable of recognizing the id-pkix-ocsp-ors-req value; and 2) be capable of identifying ORS requests according to the mechanisms described in Section 3.1.1 of this document.

```
id-pkix-ocsp           OBJECT IDENTIFIER ::= { id-ad-ocsp }
id-pkix-ocsp-ors-req   OBJECT IDENTIFIER ::= { id-pkix-ocsp X }
```

5.2 Response

The responseType field of an ORS response SHALL contain a value of id-pkix-ocsp-basic:

```
id-pkix-ocsp           OBJECT IDENTIFIER ::= { id-ad-ocsp }
id-pkix-ocsp-basic     OBJECT IDENTIFIER ::= { id-pkix-ocsp 1 }
```

The responseBytes field of an ORS response SHALL be composed of the BasicOCSPResponse syntax defined in Section 3.2.1 of this document.

5.2.1 Interpretation of CertStatus Field

The CertStatus field of BasicOCSPResponse SHALL be interpreted as follows when the value of responseType is id-pkix-ocsp-basic.

```
ORSCertStatus ::= CHOICE {
    good        [0]     IMPLICIT NULL,
```

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

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

UnknownInfo ::= NULL -- this can be replaced with an enumeration

The "good" state indicates that the certificate has not been revoked. It does not indicate that the certificate was ever issued, is or is in its validity interval.

The "revoked" state indicates that the certificate has been revoked (either permanantly or temporarily (on hold)).

The "unknown" state indicates that the responder doesn't know about the certificate being requested.

5.2.2 Response Signature Production and Validation

ORS response messages SHALL be digitally signed if and only if they contain a value for responseBytes. ORS signature calculation and production SHALL conform
to Section 3.2 of this document. The key used to sign an ORS response MAY be any key trusted by the client for this purpose. The means by which this key is established is beyond the scope of the document. Such keys include:

1. the key used to sign the subject certificate;

2. a key associated with the CA (i.e. a CA's "OCSP Signing" key);

3. a key certified by the issuing CA as an Authorized Responder; or

4. a key otherwise trusted by the requestor.

The key that signs a certificate's status information need not be the same key that signed the certificate. A certificate's issuer explicitly delegates OCSP signing authority by issuing a certificate containing a unique value for extendedKeyUsage in the OCSP signer's certificate. This certificate MUST be issued directly to the responder by the cognizant CA.

Responders MAY pre-produce signed responses. The time at which the status was known to be correct SHALL be reflected in the thisUpdate field of the response. The time at or before which newer information will be available SHALL be indicated in the nextUpdate field. The time at which the response was produced SHALL appear in the producedAt field.

5.2.3 Interpretation of Time-related Fields

The thisUpdate, nextUpdate and producedAt fields SHALL be interpreted as follows:

- thisUpdate: The time at which the status being indicated is known to be correct
- nextUpdate: The time at or before which newer information will be

available about the status of the certificate
- producedAt: The time at which the responder signed the response.

If nextUpdate is not set, the responder is indicating that newer revocation information is available all the time.

The thisUpdate and nextUpdate fields define a recommended validity interval. If CRLs are used as the basis for basic response production, this interval corresponds to the {thisUpdate, nextUpdate} interval in those CRLs; otherwise it
corresponds to equivalent database management functions.

Responses whose nextUpdate value is earlier than the local system time value SHOULD be considered unreliable. Responses whose thisUpdate time is later than the local system time SHOULD be considered unreliable. Responses where the nextUpdate value is not set are equivalent to a CRL with no time for nextUpdate.

5.2.4 Other Fields

When the byKey CHOICE of ResponderID is used, the value for KeyHash SHALL be calculated over the value of the BIT STRING subjectPublicKey key field (excluding tag and length) in the issuer's certificate.

5.2.5 CA Key Revocation

If a responder knows that a particular CA's private key has been revoked, it MAY
return the "revoked" state for all certificates issued by that CA.

6. Delegated Path Validation

This section identifies and defines the Delegated Path Validation (DPV) service.
This service may be useful to environments that wish to offload [RFC2459]-compliant certificate validation functions to a centralized server.

6.1 Request

A DPV request SHALL include a value of id-pkix-ocsp-valid-req in the requestExtensions field of TBSRequest.

```
        id-pkix-ocsp-valid-req   OBJECT IDENTIFIER ::= { id-pkix-ocsp X }
```

The corresponding value of the associated requestExtensions element SHALL contain the following:

```
DPVOptions  :: = SEQUENCE{
    pathParams              PathParameters             OPTIONAL,
    validAt           [2]   GeneralizedTime            OPTIONAL,
    revInfo           [3]   SEQUENCE OF RevocationInfo OPTIONAL,
    dPVFLags          [4]   DPVFlags                   OPTIONAL }
```

```
PathParameters ::= SEQUENCE {
    initialPolicySet  [0]   PolicyList                OPTIONAL,
    trustPoints       [1]   SEQUENCE OF ReqCert       OPTIONAL }
```

```
DPVFlags ::= BIT STRING {
    returnpath        (0),
    returnrevinfo     (1),
    returntsp         (2),
    returnpolicy      (3) }

RevocationInfo ::= CHOICE {
    cRL               CertificateList,
    oCSP              OCSPResponse }

PolicyList  ::=  SEQUENCE SIZE (1..MAX) OF OBJECT IDENTIFIER
```

The initialPolicySet option enables a requestor to establish one or more initial policy identifiers as defined in [RFC2459]. Definition of such policies is beyond the scope of this specification.

The trustPoints option enables specification of one or more certificates relevant to the relying party's trust model. If included, a successful validation request will pass through at least one of these trust points, else an "unknown" response will be generated.

A client request MAY specify the time relative to which the validation is to be performed. If omitted, the current time SHALL be assumed.

6.2 Response

A DPV response SHALL contain a value of id-pkix-ocsp-valid-rsp in the ResponseType field of the ResponseBytes syntax.

```
id-pkix-ocsp-valid-rsp    OBJECT IDENTIFIER ::= { id-pkix-ocsp X }
```

The responseBytes field of an DPV response SHALL be composed of the BasicOCSPResponse syntax defined in Section 3.2.1 of this document.

Servers that produce DPV responses SHALL execute path validation logic that produces outputs compliant with [RFC2459].

6.2.1 Interpretation of CertStatus Field

The CertStatus field of BasicOCSPResponse SHALL be interpreted as follows when the value of responseType is id-pkix-ocsp-valid.

```
DPVCertStatus ::= CHOICE {
    valid       [0]     IMPLICIT NULL,
    invalid     [1]     IMPLICIT NULL,
    unknown     [2]     IMPLICIT NULL }
```

The "valid" state SHALL be interpreted as indicating that the certificate at a minimum satisfies the path validation rules defined in [RFC2459]. Servers MAY further predicate production of a "valid" response upon additional and locally-defined authorization criteria.

The "invalid" state SHALL be interpreted as indicating that the subject certificate does not satisfy one or more conditions necessary to the production of a "valid" state.

The "unknown" state SHALL be interpreted as indicating that the server has no knowledge of the subject certificate.

If a DPV request contains a non-NULL value for initialPolicySet, all OIDs included in that set SHALL be used as initial policy identifier values in the validation logic according to [RFC2459].

If a DPV request contains a non-NULL value for trustPoints, the receiving server SHALL attempt to produce a response that incorporates these certificates. If the receiving server cannot form such a path, the server SHALL return a status value of "unknown" in the response.

6.3 Processing Considerations

[additional work needed on the various options]

7. Delegated Path Discovery

The path validation logic defined by [RFC2459] requires certificate-processing systems to accumulate the set of certificates from which certificate chains may be constructed as well as revocation data for each such certificate. These data may originate from diverse sources. Commonly used technologies for retrieving this information include X.500, LDAP, HTTP, FTP among other methods. Delegating this acquisition process to a separate server greatly simplifies and reduces the size of public-key based credential validation systems or other relying party software. It may also be useful to such software to be able to select from among various trust paths in the event multiple paths exist. The Delegated Path Discovery (DPD) service addresses these needs.

7.1 Request

A DPD request SHALL contain a value of id-pkix-ocsp-path-req in the ResponseType field of the ResponseBytes syntax.

```
id-pkix-ocsp-path-req   OBJECT IDENTIFIER ::= { id-pkix-ocsp X }
```

The corresponding value of the associated requestExtensions element SHALL contain the following:

```
DPDOptions  :: = SEQUENCE{
```

```
    retryReference          OCTET STRING,
    initialPolicySet  [0]   EXPLICIT PolicyList OPTIONAL,
    trustPoints       [1]   SEQUENCE OF ReqCert OPTIONAL,
    validAt           [2]   GeneralizedTime     OPTIONAL,
    dDPFlags          [3]   DPDFlags            OPTIONAL}
```

```
DPDFlags ::=  BIT STRING {
    returnTSP          (0),
    returnpolicy       (1),
    crlonly            (2),
    ocsponly           (3),
    either             (4) }
```

The RetryReference enables a requestor to acquire the next of potentially several valid paths known to the OCSP server based on a previous response. If this field is omitted then the request is considered to be a "new" request and the responder may return any path that meets the request criteria. If a client does specify a RetryReference then the responder SHALL NOT return any path that was previously returned with that reference (i.e. the responder MUST either return a different path meeting the request or an error).

7.2 Response

A DPD response SHALL contain a value of id-pkix-ocsp-path-rsp in the ResponseType field of the ResponseBytes syntax.

```
id-pkix-ocsp-path-rsp    OBJECT IDENTIFIER ::= { id-pkix-ocsp X }
```

The responseBytes field of a DPD response SHALL consists of the following information.

```
DPDOCSPResponse ::= SEQUENCE OF PathResponse
   -- one for each certificate included in the requestList field of TBSRequest

PathResponse ::= SEQUENCE {
      retryReference   BIT STRING,
      certificates     SEQUENCE OF Certificate,
      crls             SEQUENCE SIZE (1..MAX) OF CertificateList OPTIONAL,
      ocspReps         SEQUENCE SIZE (1..MAX) OF OCSPResponse OPTIONAL }
```

The sequence of certificates SHALL form a potentially valid certification path, in order, from end-entity certificate (element 0 of the sequence), up to and including a "final" CA certificate, (which need not, but MAY be self-certified).

The RetryReference SHOULD uniquely refer to all path validation data (including the data in the current response) that has been returned to the requester with respect to this request.

The responder MAY also include a set of CRLs and/or OCSP responses which, if included, SHOULD relate to the certificates in the set of certificates.

7.3 Processing Considerations

DPD servers SHALL:

a. Upon receipt of an authorized path discovery request, where possible, deliver to the requestor a collection of certificates and optionally CRLs and other OCSP responses that may be used to validate a certificate according to [RFC2459];

b. Either establish a stateful association enabling a requestor to serially ask for the next path via the retry option, to the extent that multiple validation paths exist and the receiving OCSP server is aware of these paths or respond with a noStateMaintained error to all retry requests if the server does not maintain state; and

c. In the event that the server is stateful and a prior response was the last path known to the responder, respond to subsequent retry requests with a noMoreData value in OSCPResponseStatus.

[additional work needed on the various options]

8. Extensions

This section defines several extensions that could be of use in specific instances. Support for any specific extension is OPTIONAL. Additional extensions MAY be defined in additional RFCs. Unrecognized extensions MUST be ignored (unless they have the critical flag set and are not understood).

8.1 Nonce

The nonce cryptographically binds a request and a response to prevent replay attacks (assuming that the requester-generated nonce is a large random number that the requester has not used previously). 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-ocsp-nonce, while the extnValue is the value of the nonce.

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

8.2 CRL References

It may be desirable for the OCSP responder to indicate the CRL on which a revoked or onHold certificate is found. This can be useful where OCSP 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-ocsp-crl, while the value will be CrlID.

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

```
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.

8.3 Acceptable Response Types

An OCSP 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-ocsp-response, and the
value AcceptableResponses. This extension is included as one of the requestExtensions in requests. The OIDs included in AcceptableResponses are the OIDs of the various response types this client can accept (e.g., id-pkix-ocsp-basic).

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

AcceptableResponses ::= SEQUENCE OF OBJECT IDENTIFIER
```

8.4 Archive Cutoff

An OCSP responder MAY retain information relevant to a certificate's validity beyond a certificate's expiration. The date obtained by subtracting this retention interval value from the producedAt time in an ORS or DPV response is defined as the certificate's "archive cutoff" date. Applications would use an archive cutoff date to contribute to a proof that a digital signature was (or was not) reliable on the date it was produced even if the certificate needed to validate the signature has long since expired. OCSP servers SHOULD include an archive cutoff date extension in responses where applicable. If included, this value SHALL be provided as an OCSP singleExtensions extension identified by id-pkix-ocsp-archive-cutoff and of syntax GeneralizedTime.

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

ArchiveCutoff ::= GeneralizedTime
```

To illustrate, if a server is operated with a 7-year retention interval policy and status was produced at time t1 then the value for ArchiveCutoff in the response would be (t1 - 7 years).

8.5 Service Locator

An OCSP server may be operated in a mode whereby the server receives a request and routes it to the OCSP server which is known to be authoritative for the

identified certificate. The serviceLocator request extension is defined for this purpose. This extension is included as one of the singleRequestExtensions in requests.

```
id-pkix-ocsp-service-locator OBJECT IDENTIFIER ::= { id-pkix-ocsp 7 }
```

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

Values for these fields are obtained from the corresponding fields in the subject certificate.

8.6 CRL Entry Extensions

All the extensions specified as CRL Entry Extensions - in Section 5.3 of [RFC2459] - are also supported as singleExtensions.

9. Security Considerations

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

A denial of service vulnerability is evident with respect to a flood of queries.
The production of a cryptographic signature significantly affects response generation cycle time, thereby exacerbating the situation. 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 (good) response is replayed prior to its expiration date but after the certificate has been revoked. Deployments of OCSP should carefully evaluate the benefit of precomputed responses against the probability of a replay attack and the costs associated with its successful execution.

Requests do not contain the responder they are directed to. This allows an attacker to replay a request to any number of OCSP responders.

The reliance of HTTP caching in some deployment scenarios may result in unexpected results if intermediate servers are incorrectly configured or are known to possess cache management faults.

Implementors are advised to take the reliability of HTTP cache mechanisms into account when deploying OCSP over HTTP.

10. Acknowledgments

The authors appreciate the hard work of all members of the IETF PKIX Working Group in refining the text of this specification. We extend special thanks to Denis Pinkas, Ambarish Malpani, and Peter Gutman for their efforts and support.

11. References

[RFC2459] Housley, R., Ford, W., Polk, W. and D. Solo, "Internet X.509 Public Key Infrastructure Certificate and CRL Profile", RFC 2459, January 1999.

[HTTP] Fielding, R., Gettys, J., Mogul, J., Frystyk, H. and T. Berners-Lee, "Hypertext Transfer Protocol -- HTTP/1.1", RFC 2068, January 1997.

[RFC2119] Bradner, S., "Key words for use in RFCs to Indicate Requirement Levels", BCP 14, RFC 2119, March 1997.

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

[X.690] ITU-T Recommendation X.690 (1994) | ISO/IEC 8825-1:1995, Information Technology - ASN.1 encoding rules: Specification of Basic Encoding Rules (BER), Canonical Encoding Rules (CER) and Distinguished Encoding Rules (DER).

[ACPROF] Farrell, S., Housley, R., "An Internet Attribute Certificate Profile for Authorization", Internet Draft draft-ietf-pkix-ac509prof-xx.txt (work in progress).

[SHA1] National Institute of Standards and Technology. FIPS Pub 180-1: Secure Hash Standard. 17 April 1995.

[RFC2437] Kaliski, B., "PKCS #1: RSA Encryption, Version 2.0", RFC 2437, October 1998.

[DSS] National Institute of Standards and Technology. FIPS Pub 186: Digital Signature Standard. 19 May 1994.

12. Authors' Addresses

Michael Myers
TraceRoute Security, Inc.
myers@coastside.net
+415.819.1362

Rich Ankney
rankney@erols.com

Carlisle Adams
Entrust Technologies
cadams@entrust.com

13. Appendix A. OCSP over HTTP

A.1 OCSP over HTTP

This section describes the formatting that will be done to the

request and response to support HTTP.

A.1.1 Request

HTTP based OCSP requests can use either the GET or the POST method to submit their requests. To enable HTTP caching, small requests (that after encoding are less than 255 bytes), MAY be submitted using GET. If HTTP caching is not important, or the request is greater than 255 bytes, the request SHOULD be submitted using POST. Where privacy is a requirement, OCSP transactions exchanged using HTTP MAY be protected using either TLS/SSL or some other lower layer protocol.

An OCSP request using the GET method is constructed as follows:

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

where {url} may be derived from the value of AuthorityInfoAccess or other local configuration of the OCSP client.

An OCSP request using the POST method is constructed as follows: The Content-Type header has the value "application/ocsp-request" while the body of the message is the binary value of the DER encoding of the OCSPRequest.

A.1.2 Response

An HTTP-based OCSP response is composed of the appropriate HTTP headers, followed by the binary value of the DER encoding of the OCSPResponse. The Content-Type header has the value "application/ocsp-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.

14. Appendix B. OCSP in ASN.1

[MODULE TO BE AGGREGATED AND IDENTIFIED]

15. Appendix C. MIME registrations

C.1 application/ocsp-request

To: ietf-types@iana.org
Subject: Registration of MIME media type application/ocsp-request

MIME media type name: application

MIME subtype name: ocsp-request

Required parameters: None

Optional parameters: None

Encoding considerations: binary

Security considerations: Carries a request for information. This request may optionally be cryptographically signed.

Interoperability considerations: None

Published specification: IETF PKIX Working Group Draft on Online Certificate Status Protocol - OCSP

Applications which use this media type: OCSP clients

Additional information:

Magic number(s): None
File extension(s): .ORQ
Macintosh File Type Code(s): none

Person & email address to contact for further information:
Ambarish Malpani <ambarish@valicert.com>

Intended usage: COMMON

Author/Change controller:
Ambarish Malpani <ambarish@valicert.com>

C.2 application/ocsp-response

To: ietf-types@iana.org
Subject: Registration of MIME media type application/ocsp-response

MIME media type name: application

MIME subtype name: ocsp-response

Required parameters: None

Optional parameters: None
Encoding considerations: binary

Security considerations: Carries a cryptographically signed response

Interoperability considerations: None

Published specification: IETF PKIX Working Group Draft on Online Certificate Status Protocol - OCSP

Applications which use this media type: OCSP servers

Additional information:

Magic number(s): None

File extension(s): .ORS
Macintosh File Type Code(s): none

Person & email address to contact for further information:
Ambarish Malpani <ambarish@valicert.com>

Intended usage: COMMON

Author/Change controller:
Ambarish Malpani ambarish@valicert.com