[MS-OCSP]:
Online Certificate Status Protocol (OCSP) Extensions

Revision Summary

Date	Revision History	Revision Class	Comments
12/18/2006	0.1		MCPP Milestone 2 Initial Availability
03/02/2007	1.0		MCPP Milestone 2
04/03/2007	1.1		Monthly release
05/11/2007	1.2		Monthly release
06/01/2007	1.2.1	Editorial	Revised and edited the technical content.

Date	Revision History	Revision Class	Comments
07/03/2007	1.2.2	Editorial	Revised and edited the technical content.
07/20/2007	1.2.3	Editorial	Revised and edited the technical content.
08/10/2007	1.2.4	Editorial	Revised and edited the technical content.
09/28/2007	1.3	Minor	Added captions to figures.
10/23/2007	1.4	Minor	Updated the technical content.
11/30/2007	2.0	Major	Updated and revised the technical content.
01/25/2008	3.0	Major	Updated and revised the technical content.

Table of Contents

1 Introduction

Online Certificate Status Protocol (OCSP) Extensions specifies the Microsoft implementation of the Online Certificate Status Protocol (OCSP) [RFC2560] and any extensions to [RFC2560].

Familiarity with **public key infrastructure (PKI)** concepts, such as asymmetric and symmetric cryptography, asymmetric and **symmetric encryption** techniques, digital **certificate** concepts, and cryptographic **key** establishment are required for a complete understanding of this protocol specification. [CRYPTO] provides an excellent introduction to cryptography and PKI concepts. [X509] provides an excellent introduction to PKI and certificate concepts.

1.1 Glossary

The following terms are defined in [MS-GLOS]:

Active Directory
Authentication Level
Authenticator
Binary Large Object (BLOB)
Certificate
Certificate Revocation Lists (CRL)
Certificate Services
Certificate Templates
Certification Authority (CA)
Client
Endpoint
Enrollment
Globally Unique Identifier (GUID)
Key
Mutual Authentication
Object Identifier (OID)
Opnum
Protocol Data Unit (PDU)
Private Key
Public Key
Public Key Algorithm
Public Key Infrastructure (PKI)
Public-Private Key Pair
Registration Authority (RA)
Remote Procedure Call (RPC)
RPC Protocol Sequence
Service Principal
Symmetric Encryption
Trust
Trust Root
Universal Naming Convention (UNC)
Universally Unique Identifier (UUID)
Well-Known Endpoint

The following terms are specific to this document:

Authentication Type: A numeric value that indicates the Security Support Provider (SSP) that is used to provide authentication, signing, and encryption for an **RPC** session.

Request: A message from a **client** to an OCSP **responder** requesting the **revocation** status of an X.509 **certificate** (see [RFC2560]).

Response: A message from an OCSP **responder** specifying the status of an X.509 **certificate** (see [RFC2560]).

Responder: A server that provides OCSP **responses** (see [RFC2560]).

Relying Party (RP): The entity (person or computer) using information from a **certificate** in order to make a security decision. Typically, the **RP** is responsible for guarding some resource and applying access control policies based on information learned from a **certificate**.

Revocation: The process of invalidating a **certificate**. For more information on revocation, see section 3.3 of [RFC3280].

Root CA: A type of **CA** that is directly **trusted** by a **relying party (RP)**. This term is not meant to imply that a **root CA** is necessarily at the top of any hierarchy; rather, the **CA** in question simply is **trusted** directly [RFC2510]. For more information, see [RFC3280].

MAY, SHOULD, MUST, SHOULD NOT, MUST NOT: These terms (in all caps) are used as described in [RFC2119]. All statements of optional behavior use either MAY, SHOULD, or SHOULD NOT.

1.2 References

1.2.1 Normative References

We conduct frequent surveys of the normative references to assure their continued availability. If you have any issue with finding a normative reference, please contact dochelp@microsoft.com. We will assist you in finding the relevant information. Please check the archive site, http://msdn2.microsoft.com/en-us/library/E4BD6494-06AD-4aed-9823-445E921C9624, as an additional source.

[FIPS140] National Institute of Standards and Technology, "Federal Information Processing Standards Publication 140-2: Security Requirements for Cryptographic Modules", December 2002, http://csrc.nist.gov/publications/fips/fips140-2/fips1402.pdf

[LWOCSP] Deacon, A. and Hurst, R., "Lightweight OCSP Profile for High Volume Environments", February 2007, http://www1.tools.ietf.org/html/draft-ietf-pkix-lightweight-ocsp-profile-09

[MS-GLOS] Microsoft Corporation, "Windows Protocols Master Glossary", March 2007.

[MS-WCCE] Microsoft Corporation, "Windows Client Certificate Enrollment Protocol Specification", June 2007.

[RFC2068] Fielding, R., Gettys, J., Mogul, J., Frystyk, H., and Berners-Lee, T., "Hypertext Transfer Protocol -- HTTP/1.1", RFC 2068, January 1997, http://www.ietf.org/rfc/rfc2068.txt

[RFC2119] Bradner, S., "Key words for use in RFCs to Indicate Requirement Levels", BCP 14, RFC 2119, March 1997, http://www.ietf.org/rfc/rfc2119.txt

[RFC2315] Kaliski, B., "PKCS #7: Cryptographic Message Syntax Version 1.5", RFC 2315, March 1998, http://www.ietf.org/rfc/rfc2315.txt

[RFC2560] Myers, M., Ankney, R., Malpani, A., Glaperin, S., and Adams, C., "X.509 Internet Public Key Infrastructure Online Certificate Status Protocol - OCSP", RFC 2560, June 1999, http://www.ietf.org/rfc/rfc2560.txt

[RFC2616] Fielding, R., et al., "Hypertext Transfer Protocol -- HTTP/1.1", RFC 2616, June 1999, http://www.ietf.org/rfc/rfc2616.txt

[RFC2797] Myers, M., Liu, X., Schaad, J., and Weinstein, J., "Certificate Management Messages Over CMS", RFC 2797, April 2000, http://www.ietf.org/rfc/rfc2797.txt

[RFC2986] Nystrom, M. and Kaliski, B., "PKCS#10: Certificate Request Syntax Specification", RFC 2986, November 2000, http://www.ietf.org/rfc/rfc2986.txt

[RFC3280] Housley, R., Polk, W., Ford, W., and Solo, D., "Internet X.509 Public Key Infrastructure Certificate and Certificate Revocation List (CRL) Profile", RFC 3280, April 2002, http://www.ietf.org/rfc/rfc3280.txt

[X509] ITU-T, "Information Technology - Open Systems Interconnection - The Directory: Public-Key and Attribute Certificate Frameworks", Recommendation X.509, August 2005, http://www.itu.int/rec/T-REC-X.509/en

Note There is a charge to download the specification.

[X660] ITU-T, "Information Technology - Open Systems Interconnection - Procedures for the Operation of OSI Registration Authorities: General Procedures and Top Arcs of the ASN.1 Object Identifier Tree", Recommendation X.660, August 2004, http://www.itu.int/rec/T-REC-X.660/en

Note There is a charge to download the specification.

[ITUX690] ITU-T, "ASN.1 Encoding Rules: Specification of Basic Encoding Rules (BER), Canonical Encoding Rules (CER) and Distinguished Encoding Rules (DER)", Recommendation X.690, July 2002, http://www.itu.int/ITU-T/studygroups/com17/languages/X.690-0207.pdf

1.2.2 Informative References

[CRYPTO] Menezes, A., Vanstone, S., and Oorschot, P., "Handbook of Applied Cryptography", 1997, http://www.cacr.math.uwaterloo.ca/hac/

[HOWARD] Howard, M., "Writing Secure Code", Microsoft Press, 2002, ISBN: 0735617228.

1.3 Protocol Overview (Synopsis)

The Online Certificate Status Protocol (OCSP) [RFC2560] provides a mechanism, in lieu of or as a supplement to checking against a periodic **CRL**, to obtain timely information regarding the **revocation** status of a certificate ([RFC3280] section 3.3). The OCSP Protocol ([RFC2560]) enables applications to determine the (Revocation) state of an identified X.509 certificate [X509]. This protocol specifies the data that needs to be exchanged between an application checking the status of a certificate and the server providing that status.

OCSP is a component of a public key infrastructure (PKI). A PKI consists of a system of digital certificates, **certification authorities (CAs)**, and other **registration authorities (RAs)** that verify and authenticate the validity of each party involved in an electronic transaction through the use of **public key** cryptography.

The certificate status received as a result of using OCSP is known as a **response** from an OCSP **responder**. The OCSP request/response process involves a number of different machines (or functions that might be hosted on the same machine), as indicated in Figure 1.

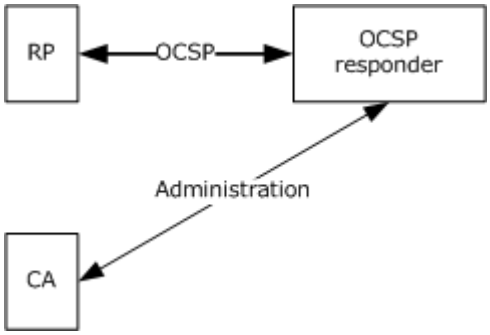

Figure 1: Response from an OCSP

In Figure 1 the principal components are:

1. CA: The CA that provides certificate status information to the OCSP responder through the use of CRLs.
2. **Relying party (RP)**: The resource guard that validates a certificate chain and contacts an OCSP responder to **request** certificate status.
3. OCSP responder: An authoritative source for certificate revocation status (see [RFC3280] section 3.3). The protocols and data structures used for OCSP are defined in section 2.2. The connection over which OCSP is conducted is shown in Figure 1 as a solid bold horizontal line.

This document specifies the data structures and messages that constitute an OCSP request and an OCSP response.

1.4 Relationship to Other Protocols

The Hypertext Transfer Protocol (HTTP/1.1) [RFC2616] MUST be the transport protocol for Online Certificate Status Protocol (OCSP) Extensions messages.

1.5 Prerequisites/Preconditions

OCSP requires the Hypertext Transfer Protocol 1.1 (HTTP/1.1) [RFC2616]) for transport of all messages.

OCSP assumes the following:

The **client** MAY discover the OCSP server through the AIA extension defined in [RFC3280] section 4.2.2.1. or through a URL configured through out-of-band means.<1>

1.6 Applicability Statement

This protocol is applicable to an environment that wants to enable clients to interact with an OCSP responder for the purpose of requesting the revocation status of an [X509] certificate.

1.7 Versioning and Capability Negotiation

None.

1.8 Vendor-Extensible Fields

None.

1.9 Standards Assignments

None.

2 Messages

The following sections specify how OCSP Extensions messages are transported and encoded on the wire.

2.1 Transport

OCSP is commonly used over HTTP [RFC2068], although additional transports are allowed per [RFC2560] section 4.1.<2>

2.2 Message Syntax

The following sections define the message syntax for OCSP Extensions. OCSP messages are defined in ASN.1 as described in [X660] and encoded by using DER encoding as described in [ITUX690].

2.2.1 Common Structures

Clients and servers that implement OCSP MUST use the ASN.1 structures specified in [RFC2560] when constructing an OCSP request and response. The following fields are introduced and defined in section 4 of [RFC2560] and are used by OCSP:

```
OCSPRequest
    TBSRequest
    OPTIONAL Signature

OCSPResponse
    OCSPResponseStatus
    ResponseBytes
```

Detailed server processing information is in section 3.2

3 Protocol Details

The following sections specify protocol details, including abstract data models and message processing rules.

3.1 Client Details

The client role in OCSP Extensions is to generate a request, as specified in section 2.2.1, and upon receipt validate the response.

3.1.1 Abstract Data Model

None.

3.1.2 Timers

None.

3.1.3 Initialization

None.

3.1.4 Higher-Layer Triggered Events

None.

3.1.5 Message Processing Events and Sequencing Rules

OCSP request creation MUST adhere to [RFC2560] section 4.1.<3>

3.1.6 Timer Events

None.

3.1.7 Other Local Events

None.

3.2 Server Details

The following sections define the server sequencing and processing rules for the OCSP implementation.

3.2.1 Abstract Data Model

The server MUST maintain a list of revoked certificates and maintain the following fields for each revoked certificate:

- Certificate serial number, as specified in [RFC3280] section 4.1.2.2.
- Revocation date and time, as specified in [RFC3280] section 5.3.3.
- Revocation reason, as specified in [RFC3280] section 5.3.1.

It MUST maintain a **private key** with which to sign OCSP responses. It MUST hold a certificate on the associated public key, which is delivered to OCSP clients to verify its authority to sign OCSP responses.

3.2.2 Timers

None.

3.2.3 Initialization

The server MUST acquire a certificate as defined in [RFC2560] section 4.2.2.2.

3.2.4 Higher-Layer Triggered Events

None.

3.2.5 Message Processing Events and Sequencing Rules

OCSP request processing and response generation MUST adhere to [RFC2560] section 4.2.<4>

3.2.6 Timer Events

None.

3.2.7 Other Local Events

None.

4 Protocol Example

The client determines that it requires validating the revocation status of a certificate. Once the client invokes the revocation checking process the following sequence of events occur:

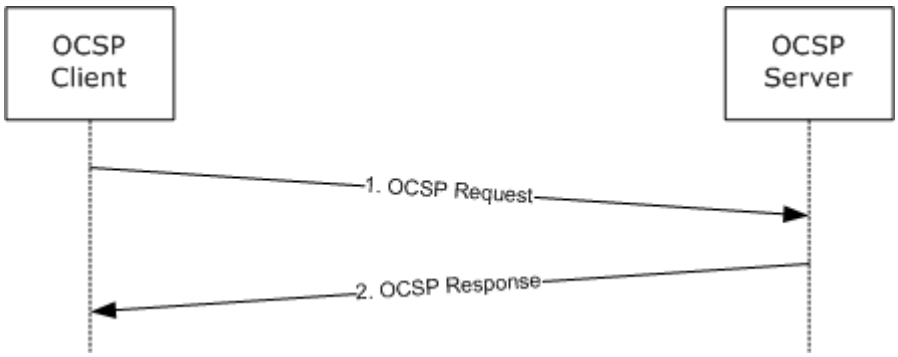

Figure 2: Revocation checking process

1. The OCSP client generates an OCSP request as specified in section 3.1.5 and submits the request to the responder.

2. The responder inspects the requests and generates a response as specified in section 3.2.5.

5 Security

The following sections specify security considerations for implementers of OCSP Extensions.

5.1 Security Considerations for Implementers

Any cryptographic protocol has security considerations dealing with key handling during cryptographic operations and key distribution. Although a public-key certificate is not a protocol by itself, it has most of the same security considerations of a cryptographic protocol in the sense that a public key certificate is a message from the CA to the RP—a message addressed, in effect, "to whom it may concern." A cryptographic protocol that deals with the transmission or issuance or other use of a public key certificate therefore has security considerations in two areas: around the protocol itself and around the certificate and its use.

In addition, a certificate binds two or more pieces of information together. In the most common case, that would be a public key and a name. The name in such a certificate has security relevance and there are security considerations around the use and provisioning of those names. In some certificate forms, there are attributes bound to either a name or a key, and there are security considerations around the use and provisioning of those attributes.

5.1.1 Keeping Information Secret

Any cryptographic key must be kept secret. One must also keep secret any function of a secret (such as a key schedule), as knowing such functions would reduce an attacker's work in cryptanalyzing the secret.

When a secret must be in the normal memory of a general purpose computer in order to be used, that secret should be erased (for example, replaced with a constant value, like 0) as soon possible after use.

A secret may be kept in a specially protected memory where it can be used without being erased. Typically, one finds such memory in a Hardware Security Module (HSM). If an HSM is used, it should be compliant with [FIPS140], or the equivalent at a level consistent with the security requirements of the customer deploying the cryptographic protocol or the CA that uses the HSM.

5.1.2 Coding Practices

Any implementation of a protocol exposes code to inputs from attackers. Such code must be developed according to secure coding and development practices in order to avoid buffer overflows, denial-of-service attacks, escalation of privilege, and disclosure of information. For an introduction to these concepts, as well as secure development best practices and common errors, see [HOWARD].

5.1.3 Security Consideration Citations

Implementers of this protocol should take care to consider the following security considerations:

- A client or server should follow generally accepted principles of secure key management. For more information, see section 9 of [RFC3280]. For an introduction to these generally accepted principles, see [CRYPTO] and [HOWARD].
- Clients and servers should validate cryptographic parameters prior to issuing or accepting certificates. For more information, see section 9 of [RFC2797].

- A client and server should validate and verify the certificate path information identified in section 6 of [RFC3280]. See section 9 of [RFC3280] for more information on the requirement for Certificate path validation.
- A client and server should validate and verify the freshness of revocation information of all digital certificates prior to usage, **trust**, or encryption as identified in section 6.3 of [RFC3280]. See section 9 of [RFC3280] for more information on the requirement for revocation freshness.
- A client or server should follow all security considerations in section 5 of [RFC2560].
- A client or server should follow all security considerations discussed throughout [RFC2315] and [RFC2986] as neither normative reference has a specific security section.
- A client and server should use an authenticated HTTP session between client and server to mitigate denial of service attacks. For more information on generic denial of service mitigation techniques, see [HOWARD].

5.2 Index of Security Parameters

None.

6 Appendix A: Windows Behavior

The information in this specification is applicable to the following versions of Windows:

- Windows Vista
- Windows Server 2008

Exceptions, if any, are noted below. Unless otherwise specified, any statement of optional behavior in this specification prescribed using the terms SHOULD or SHOULD NOT implies Windows behavior in accordance with the SHOULD or SHOULD NOT prescription. Unless otherwise specified, the term MAY implies that Windows does not follow the prescription.

<1> Section 1.5: Windows Vista uses only the URL specified in the validated certificate AIA extension.

<2> Section 2.1: OCSP Extensions is conformant with OCSP over HTTP as specified in [RFC2560] Appendix A.

<3> Section 3.1.5: OCSP clients running Windows Vista generate the OCSP request as follows:

1. The version field is set to 1.
2. The OPTIONAL **requestorName** and **requestExtensions** request fields are not included in the request.
3. The requestList always contains only one request.
4. The **CertId** field always uses the SHA-1 hash algorithm.
5. The OCSP Extensions client does not sign the requests.

When processing the response from the server, an OCSP Extensions client enforces that the response is signed by one of the following keys:

1. The private key that was used to sign the Certificate inspected.
2. A private key with a corresponding Certificate that was signed using the same private key that was used to sign the Certificate inspected.

<4> Section 3.2.5: The Microsoft OCSP Extensions server processes the OCSP requests and generates the OCSP response as follows:

1. If a **requestList** field of the request includes multiple requests, the OCSP Extensions responder will reject the request with HTTP (see [RFC2616]) error 500 for failure.
2. If the request is signed, the OCSP Extensions responder will reject the request with HTTP (see [RFC2616]) error 500 for failure.
3. The **responseType** field is always id-pkix-ocsp-basic as defined in [RFC2560] section 4.2.1.
4. The responses field of the OCSP response will always include a single response.
5. Nonce extension defined in [RFC2560] section 4.4.1 can be included, based on the OCSP responder policy, in the **responseExtensions** field of the response. No other extensions are supported.

6. The OCSP Extensions responder includes a non-critical extension with **OID** (1.3.6.1.4.1.311.21.4) only if the CA issues a CRL that contains the same extension. The extension value contains the time when the next OCSPResponse (for the specific request) is expected to be published. This may be sooner than the **NextUpdate** field. The extension value is DER-encoded and is defined in ASN.1 [X509] as:

```
CHOICE {
    utcTime    UTCTime,
    generalTime    GeneralizedTime
}
```

7. The OCSP Extensions responder adds the HTTP headers as specified in [LWOCSP] for an OCSPResponse.

8. If the OCSPRequest is preceded by the conditional HTTP headers "If-Modified-Since" or "If-None-Match" [RFC2068], the OCSP Extensions responder evaluates if it has a newer OCSPResponse (newer than specified in the condition) for the OCSPRequest, and responds with an HTTP 304 (not modified) status if it does not (see [RFC2068]).

7 Index

T

V

W