

Online Certificate Status Protocol Mobile Profile

Candidate Version V1.0 – 27 Jan 2004

Open Mobile Alliance
OMA-WAP-OCSP-V1_0-20040127-C

Continues the Technical Activities
Originated in the WAP Forum



Use of this document is subject to all of the terms and conditions of the Use Agreement located at <http://www.openmobilealliance.org/UseAgreement.html>.

Unless this document is clearly designated as an approved specification, this document is a work in process, is not an approved Open Mobile Alliance™ specification, and is subject to revision or removal without notice.

You may use this document or any part of the document for internal or educational purposes only, provided you do not modify, edit or take out of context the information in this document in any manner. Information contained in this document may be used, at your sole risk, for any purposes. You may not use this document in any other manner without the prior written permission of the Open Mobile Alliance. The Open Mobile Alliance authorizes you to copy this document, provided that you retain all copyright and other proprietary notices contained in the original materials on any copies of the materials and that you comply strictly with these terms. This copyright permission does not constitute an endorsement of the products or services. The Open Mobile Alliance assumes no responsibility for errors or omissions in this document.

Each Open Mobile Alliance member has agreed to use reasonable endeavors to inform the Open Mobile Alliance in a timely manner of Essential IPR as it becomes aware that the Essential IPR is related to the prepared or published specification. However, the members do not have an obligation to conduct IPR searches. The declared Essential IPR is publicly available to members and non-members of the Open Mobile Alliance and may be found on the “OMA IPR Declarations” list at <http://www.openmobilealliance.org/ipr.html>. The Open Mobile Alliance has not conducted an independent IPR review of this document and the information contained herein, and makes no representations or warranties regarding third party IPR, including without limitation patents, copyrights or trade secret rights. This document may contain inventions for which you must obtain licenses from third parties before making, using or selling the inventions. Defined terms above are set forth in the schedule to the Open Mobile Alliance Application Form.

NO REPRESENTATIONS OR WARRANTIES (WHETHER EXPRESS OR IMPLIED) ARE MADE BY THE OPEN MOBILE ALLIANCE OR ANY OPEN MOBILE ALLIANCE MEMBER OR ITS AFFILIATES REGARDING ANY OF THE IPR'S REPRESENTED ON THE “OMA IPR DECLARATIONS” LIST, INCLUDING, BUT NOT LIMITED TO THE ACCURACY, COMPLETENESS, VALIDITY OR RELEVANCE OF THE INFORMATION OR WHETHER OR NOT SUCH RIGHTS ARE ESSENTIAL OR NON-ESSENTIAL.

THE OPEN MOBILE ALLIANCE IS NOT LIABLE FOR AND HEREBY DISCLAIMS ANY DIRECT, INDIRECT, PUNITIVE, SPECIAL, INCIDENTAL, CONSEQUENTIAL, OR EXEMPLARY DAMAGES ARISING OUT OF OR IN CONNECTION WITH THE USE OF DOCUMENTS AND THE INFORMATION CONTAINED IN THE DOCUMENTS.

© 2004 Open Mobile Alliance Ltd. All Rights Reserved.

Used with the permission of the Open Mobile Alliance Ltd. under the terms set forth above.

Contents

1. SCOPE	4
2. REFERENCES	5
2.1 NORMATIVE REFERENCES	5
2.2 INFORMATIVE REFERENCES	5
3. TERMINOLOGY AND CONVENTIONS	6
3.1 CONVENTIONS	6
3.2 DEFINITIONS	6
3.3 ABBREVIATIONS	6
4. INTRODUCTION	7
5. OCSP PROFILE AND USE	8
5.1 COMPATIBILITY WITH EXISTING INFRASTRUCTURE	8
5.2 OCSP REQUESTS	8
5.3 OCSP RESPONSES	8
5.4 CLIENT BEHAVIOR	9
5.4.1 General.....	9
5.4.2 OCSP response status processing	9
5.4.3 Client behavior if no response is received in a set time	9
5.5 IDENTIFYING, LOCATING AND COMMUNICATING WITH AN OCSP RESPONDER	9
5.5.1 Responder identification and location.....	9
5.5.2 OCSP transport protocol requirements	9
6. SECURITY CONSIDERATIONS	11
6.1 ACTIVE ATTACKS	11
6.1.1 Replay attacks	11
6.1.2 Man-in-the-middle attacks	11
6.1.3 Impersonation attacks	11
6.1.4 Denial of service attacks	11
6.2 PASSIVE ATTACKS	11
6.3 OTHER CONSIDERATIONS	11
APPENDIX A. STATIC CONFORMANCE REQUIREMENTS (NORMATIVE)	12
APPENDIX B. CHANGE HISTORY (INFORMATIVE)	15
B.1 APPROVED VERSION HISTORY	15
B.2 DRAFT/CANDIDATE VERSION 1.2 HISTORY	15
APPENDIX C. EXAMPLE OF OCSP MESSAGES	16

1. Scope

Open Mobile Alliance (OMA) is a consortium of mobile industry companies working to define an industry-wide specification for developing applications that operate over wireless communication networks. The scope for the OMA is to define a set of specifications to be used by service applications. The wireless market is growing very quickly and reaching new customers and services. To enable operators and manufacturers to meet the challenges in advanced services, differentiation and fast/flexible service creation, OMA defines a set of protocols in transport, session and application layers. For additional information on the Wireless Application Environment, refer to [WAESPEC].

This specification defines a profile of the Online Certificate Status Protocol (OCSP) [RFC 2560] for mobile environments. OCSP defines a protocol used to determine the current status of a digital certificate in lieu of using standard Certificate Revocation Lists (CRL's). The scope of this work is to profile OCSP in such a way to ensure it can be used efficiently by limited wireless devices and to be interoperable with OCSP responders already available in the market. This profile is defined as an OMA service enabler to ensure its use with any existing and future OMA applications and/or services.

2. References

2.1 Normative References

- [IOPProc] “OMA Interoperability Policy and Process”. Open Mobile Alliance™, OMA-IOP-Process-V1_1. <http://www.openmobilealliance.org>
- [RFC2119] “Key words for use in RFCs to Indicate Requirement Levels”. S. Bradner. March 1997. URL:<http://www.ietf.org/rfc/rfc2119.txt>
- [RFC2234] “Augmented BNF for Syntax Specifications: ABNF”. D. Crocker, Ed., P. Overell. November 1997. <http://www.ietf.org/rfc/rfc2234.txt>
- [TLS] “WAP TLS profile and Tunneling”. WAP Forum™. <http://www.openmobilealliance.org>
- [WAP HTTP] “Wireless profiled HTTP”. WAP Forum™. <http://www.openmobilealliance.org>
- [WAPCert] “WAP Cert Profile”. WAP Forum™, WAP-211-WAPCerta. <http://www.openmobilealliance.org>
- [RFC 3280] “Internet X.509 Public Key Infrastructure – Certificate and CRL Profile,” Housley, R., Ford, W., Polk, W., and D.Solo, IETF RFC 3280, April 2002. <http://www.ietf.org/rfc/rfc3280.txt>
- [RFC 2560] “X.509 Internet Public Key Infrastructure - Online Certificate Status Protocol - OCSP,” M. Myers, et al., IETF RFC 2560, June 1999. URL: <http://www.ietf.org/rfc/rfc2560.txt>

2.2 Informative References

- [WAPARCH] “WAP Architecture”. Open Mobile Alliance™. WAP-210-WAPArch. <http://www.openmobilealliance.org>
- [ISO/IEC 8824-1] ISO/IEC 8824-1, "Information Technology - Abstract Syntax Notation One (ASN.1): Specification of basic notation," International Organization for Standardization, 1998.
- [ISO/IEC 8825-1] ISO/IEC 8825-1, "Information technology - ASN.1 encoding rules: Specification of Basic Encoding Rules (BER), Canonical Encoding Rules (CER) and Distinguished Encoding Rules (DER)," International Organization for Standardization, 1998.
- [ISO/IEC 9594-8] ISO/IEC 9594-8, "Information technology - Open systems interconnection – The Directory: Authentication Framework," International Organization for Standardization, 1998.

3. Terminology and Conventions

3.1 Conventions

The key words “MUST”, “MUST NOT”, “REQUIRED”, “SHALL”, “SHALL NOT”, “SHOULD”, “SHOULD NOT”, “RECOMMENDED”, “MAY”, and “OPTIONAL” in this document are to be interpreted as described in [RFC2119].

All sections and appendixes, except “Scope” and “Introduction”, are normative, unless they are explicitly indicated to be informative.

3.2 Definitions

Client - A device (or application) that initiates a request for a connection with an OCSP server

Server - A device (or application) that passively waits for OCSP requests from one or more clients. A server may accept or reject a connection request from a client.

3.3 Abbreviations

ASN.1	Abstract Syntax Notation 1, as defined in [ISO/IEC 8824-1]
CA	Certification Authority
CRL	Certificate Revocation List
DER	Distinguished Encoding Rules, as defined in [ISO/IEC 8825-1]
HTTP	Hypertext Transfer Protocol, as defined in [WAP HTTP]
OCSP	Online Certificate Status Protocol, as defined in [RFC2560]
OMA	Open Mobile Alliance
RSA	Rivest-Shamir-Adleman public key algorithm
SHA-1	Secure Hash Algorithm
TLS	Transport Layer Security Protocol, as defined in [TLS]
WAP	Wireless Application Protocol

4. Introduction

Several specifications defined by OMA require the use of digital certificates as defined by [ISO/IEC 9594-8] and [RFC 3280] and further profiled by [WAPCert]. While digital certificates provide a secure mechanism to identify and authenticate an entity via the verification of a digital signature, there must also be mechanisms in place to validate that the certificate, and the associated private key, in use is in fact still considered trusted and valid. This issue is commonly known as certificate validation and is based on the concept of certificate revocation. Certificates may be revoked for various reasons, including, but not limited to, change of name, change of association between subject and CA and compromise or suspicion of compromise of the corresponding private key. There must be mechanisms in place for a requesting entity to retrieve status information of a certificate that may have been revoked. One method, as defined by [ISO/IEC 9594-8] and [RFC3280] defines the concept of certificate revocation lists (CRLs) that are used to convey information about certificates revoked by a Certification Authority (CA). Although CRL's are in use in many environments, they have some properties that make their use in mobile environments unattractive. In particular, the use of CRL's requires additional processing by the client that may be difficult in a constrained mobile device. Also, CRL's are often quite large in size and thus raise network latency and bandwidth issues.

The Internet Engineering Task Force's Public Key Infrastructure working group has defined an alternative method of certificate validation that does not rely on CRL's. The Online Certificate Status Protocol (OCSP) as defined by [RFC 2560] replaces the CRL concept with a simple certificate status request and response protocol to a central server. This server, also known as an OCSP responder, is authorized to respond with certificate status information.

To perform an OCSP certificate status check, a client sends a request to an OCSP responder. The OCSP responder then determines the revocation status of the requested certificate and constructs the corresponding OCSP response. This response is typically signed by the OCSP responder to ensure data integrity and that the response originated from an authoritative source.

While OCSP requests and responses normally will be quite limited in size, there is a potential (e.g. through the use of unbounded lists and extensions) for them to become large and complex. Some OCSP responders may also require clients to authenticate all their requests. These possibilities are not normally a major concern in the desktop and portable PC environment, where multiple megabytes of memory and high-speed network connections are available. Mobile devices, on the other hand, typically have limited CPU power and memory and a wireless network interface of relatively low bandwidth. For these devices, generating, signing, receiving and validating certificate status messages may be problematic. If secure messaging and Internet e-commerce are to be extended into today's wireless devices, support for certificate status checking by relying parties using mobile clients is needed. A profile of OCSP that is tuned for the mobile environment is therefore desirable. This specification seeks to meet this need.

It is assumed the reader of this document has knowledge of [RFC 2560].

5. OCSP Profile and Use

5.1 Compatibility with existing infrastructure

To the greatest extent possible, OCSP entities conformant with this specification should work interchangeably with other Internet based OCSP clients and servers in order to leverage the existing infrastructure. In accordance with this, OCSP clients conformant with this specification (henceforth: "Clients") MUST support all mandatory features of [RFC 2560] unless specified otherwise in the following.

5.2 OCSP requests

Clients MUST be able to generate **OCSPRequest** values as follows:

- Clients MAY sign **OCSPRequests**.
Note: Signed requests increase the complexity of client implementations and increase the size of transmitted client requests. However signed requests might be required by some environments where authentication of the client is required by the OCSP responder.
- If the **OCSPRequest** is signed, the client SHALL specify its name in the **requestorName** field, otherwise Clients SHOULD NOT include the **requestorName** field in the **OCSPRequest**. OCSP servers must however be prepared to receive unsigned OCSP requests that contains the **requestorName** field, but must realize that the provided value is not authenticated.
- Clients MAY include the **nonce** extension in the **OCSPRequest**.
- Clients SHOULD NOT send more than one **Request** in each **OCSPRequest.requestList**
- Clients MUST use the SHA-1 algorithm when calculating **CertID** values. Clients MUST NOT truncate calculated values.

Note: To avoid needless network traffic, this profile recommends applications to verify the signature of signed data before checking the status of certificates used to verify the data. If the signature is invalid or the client is not able to verify it, an OCSP check is not required.

5.3 OCSP responses

This profile constrains clients' support of possible **OCSPResponse** values as follows:

- Clients MUST be able to parse and accept **BasicOCSPResponse** values as identified by the **id-pkix-ocsp-basic** object identifier.
- Clients MUST have the ability to verify signatures computed with the **sha1WithRSAEncryption** signature algorithm. Clients MAY support additional algorithms as specified in [RFC2560].
- Clients MUST be able to accept **Responses** values that contain as many **SingleResponse** values as they are capable of sending **Requests** in an **OCSPRequest.requestList**.
- Clients that have the ability to include a nonce (**id-pkix-ocsp-nonce**) in the request MUST be able to parse the nonce extension in a received **BasicOCSPResponses** and, when applicable, verify that the sent and received nonces are the same.

Note: See Section 5.4.1 for additional details regarding the processing of nonces.

- Clients MUST NOT fail OCSP response processing due to unknown non-critical extensions in a **BasicOCSPResponse**.
- Clients MUST support handling of OCSP responses of the size up to 3000 bytes.

5.4 Client behavior

5.4.1 General

To ensure freshness OCSP clients **MUST NOT** accept a response that is out dated (see Section 6.1.1). Failing this, the client **MUST** inform the calling application that the certificate status is unknown.

A client, which cannot determine the freshness of an **OCSPResponse** by means other than using OCSP nonces **MUST NOT** accept an OCSP response that does not contain a nonce matching a nonce sent in the corresponding **OCSPRequest**.

Clients **MUST** validate the signature on a fresh response.

5.4.2 OCSP response status processing

OCSP response status **good**: The client **MUST** inform the calling application that the certificate has not been revoked.

The client **SHOULD NOT** accept an OCSP response that indicates (in the **nextUpdate** field) that a newer response is available.

An **OCSPResponse** **MAY** be cached by the client for future use if the **ResponseData.nextUpdate** field is present in the received **OCSPResponse**, and the value is later than the client's current time. The **OCSPResponse** **SHALL NOT** be cached beyond the point at which newer certificate status information is available. See section 6.3.

OCSP response status **revoked**: The client **MUST** inform the calling application that the signature is untrusted and abort any further processing of the signed data.

OCSP response status **unknown**: The client **MUST** inform the calling application about the unknown certificate status. This profile **RECOMMENDS** calling applications to warn the user about the unknown certificate status and give the user the option to continue or abort the processing of the data, with a default option of abort.

5.4.3 Client behavior if no response is received in a set time

Clients **MUST** have a set and documented time period before timing out waiting for an OCSP response. Clients that do not receive a response within their time-out period **SHOULD** retry the OCSP certificate status check by sending another **OCSPRequest**. Clients **MUST** limit the number of retry attempts.

Note: A round-trip time of 4 seconds is an estimated worst case for OCSP over a 9.6 kb/s network link. It is assumed that networks will have a bandwidth of 9.6 kb/s or more.

5.5 Identifying, locating and communicating with an OCSP responder

5.5.1 Responder identification and location

Clients **MUST** support OCSP delegation, i.e. recognize the use of the **id-kp-OCSPSigning** object identifier value in an OCSP responder's certificate.

Clients **MUST** support the **authorityInfoAccess** extension as defined in [RFC 3280], and **MUST** recognize the **id-ad-ocsp** access method. This enables CAs to inform clients how they can contact the OCSP service.

5.5.2 OCSP transport protocol requirements

Clients **MUST** support OCSP over wireless-profiled HTTP [WAP HTTP]. Clients **SHOULD** support the use of wireless profiled TLS [TLS] when carrying out OCSP over HTTP.

When contacting an OCSP server over HTTP,

- clients **MUST** use the GET method (to enable for OCSP response caching) as described in Appendix A.1 of RFC2560 [RFC 2560] when sending requests that are less than 255 bytes.
- clients **SHOULD** use the POST method as described in Appendix A.1 of RFC2560-RFC2560] when sending requests larger than 255 bytes
- clients **MUST** use base64 encoding (without any CR or LF characters) for the **OCSPRequest** message and append it to the given URI
- clients **MUST** url-encode the base64-encoded **OCSPRequest**, e.g.

<http://ocsp.example.com/MEowSDBGMEQwQjAKBggqhkjG9w0CBQQQ7sp6GTKpL2dAdeGaW267owQQqInESWQD0mGeBArSgv%2FBWQIQLJx%2Fg9xF8oySYzol80Mbpq%3D%3D>

- in response to a properly formatted **OCSPRequest**, the OCSP server shall include the binary value of the DER encoding of the **OCSPResponse** as the message body, preceded by (at least) the following HTTP headers:

```
Content-Type: application/ocsp-response
Content-Transfer-Encoding: binary
Content-Length: <OCSP response length>
```

```
<binary OCSPResponse>
```

6. Security Considerations

This section is informative.

6.1 Active Attacks

6.1.1 Replay attacks

The use of nonces provides a mean for clients to ensure that a received OCSP response is fresh. Clients not using nonces can (see Section 5.4.1) still limit the opportunity for possible replay attacks by ensuring that any received response is reasonably fresh (subject to local policy).

Note: Methods to maintain a secure and synchronized time is outside of the scope for this specification.

6.1.2 Man-in-the-middle attacks

To prevent this class of attack, the client must properly validate the signature on the response, as defined in Section 5.4.1. The use of signed responses in OCSP serves the purpose to authenticate the identity of the OCSP responder that has authority to sign request on the CA's behalf. As described in Section 5.5, clients must ensure that the **id-kp-OCSPSigning** extendedKeyUsage bit is present in the certificate used to sign the response.

6.1.3 Impersonation attacks

The use of signed responses in OCSP serves the purpose to authenticate the identity of OCSP Responder. As defined in Section 5.4.1 clients must properly validate the signature of the OCSP response to ensure an authorized responder created it.

6.1.4 Denial of service attacks

Protection against denial of service attacks depends on the implementation of OCSP server, and thus outside the scope of this profile.

6.2 Passive attacks

Passive attacks are generally not considered a threat in an OCSP environment. Traffic analysis may however reveal usage patterns and transport layer security may be used whenever this is a concern.

6.3 Other considerations

Clients that do not have the capability to synchronise its date and time to a reliable source are reliant on the user to accurately set the date and time. An inaccurate date and time may lead to the acceptance of revoked certificates or the rejection of valid certificates.

All the security considerations in [RFC 2560] apply to this profile.

Appendix A. Static Conformance Requirements (Normative)

The notation used in this appendix is specified in [IOPProc].

Item	Function	Reference	Status	Requirement
OCSP-C-001	Support of mandated features in RFC2560	5.1	M	
OCSP-C-002	Support for OCSPRequest	5.2	M	
OCSP-C-003	Support for requestorName field in OCSPRequest	5.2	O	
OCSP-C-003a	Clients MAY sign OCSPRequests	5.2	O	
OCSP-C-003b	If OCSPRequests are not signed then the client is not required to include the requestorName field in the OCSPRequest	5.2	O	
OCSP-C-004	Support for the use of nonce extension in OCSPRequest	5.2	O	
OCSP-C-005	Support for sending more than one Request in an OCSPRequest.requestList	5.2	O	
OCSP-C-006	Use of SHA-1 hashing algorithm for calculating CertID values.	5.2	M	
OCSP-C-007	Exclude use of truncated values	5.2	M	
OCSP-C-009	Support for parsing of BasicOCSPResponse	5.3	M	
OCSP-C-011	Verification of signatures with sha1WithRSAEncryption	5.3	M	
OCSP-C-011a	Support for additional algorithms	5.3	O	
OCSP-C-012	Clients MUST be able to accept Responses values that contain as many SingleResponse values as they are capable of sending Requests in an OCSPRequest.requestList	5.3	M	
OCSP-C-013	Client parsing of nonce extensions	5.3	M	
OCSP-C-015	Client Failure due to critical extensions	5.3	M	
OCSP-C-016	Support for OCSP response size upto 3000	5.3	M	

Item	Function	Reference	Status	Requirement
	bytes			
OCSP-C-017	OCSP clients MUST NOT accept a response that is out dated	5.4.1	M	
OCSP-C-019	A client which cannot determine the freshness of an OCSP response by means other than using OCSP nonces Clients MUST NOT accept an OCSP response that does not contain a nonce matching a nonce sent in the corresponding OCSPRequest .	5.4.1	M	
OCSP-C-020	Client MUST inform the calling application that the certificate status is unknown	5.4.1	M	
OCSP-C-021	Client validation of the signature on a fresh response	5.4.1	M	
OCSP-C-022	Responses that have the status good	5.4.2	M	
OCSP-C-022a	Should not accept a response that indicates that a newer response is available	5.4.2	M	
OCSP-C-022b	Client MAY cache an OCSP response	5.4.2	O	
OCSP-C-022c	Client shall not cache a stale OCSPResponse	5.4.2	M	
OCSP-C-023	Responses that have the status revoked	5.4.2	M	
OCSP-C-024	Responses that have the status unknown	5.4.2	O	
OCSP-C-024a	Warn user of unknown certificate status	5.4.2	M	
OCSP-C-025	Client set time period before timing out waiting for an OCSP Response	5.4.3	M	
OCSP-C-026	Retry of OCSP status check	5.4.3	O	
OCSP-C-027	Limiting the number of OCSPRequest retry attempts.	5.4.3	M	
OCSP-C-028	Client support for OCSP delegation	5.5.1	M	
OCSP-C-029	Client support for authorityInfoAccess	5.5.1	M	

Item	Function	Reference	Status	Requirement
	extension			
OCSP-C-030	Recognition of id-ad-ocsp access method	5.5.1	M	
OCSP-C-031	Client support of OCSP over wireless profiled HTTP	5.5.2	M	
OCSP-C-032	Client support of OCSP over wireless profiled TLS	5.5.2	O	
OCSP-C-033	Support for use of the GET method to allow for OCSP response caching	5.5.2	M	
OCSP-C-034	Support for use of the POST method to allow for OCSP response larger than 255 bytes	5.5.2	O	
OCSP-C-035	Use of base64 encoding	5.5.2	M	
OCSP-C-037	Use of url-encoding	5.5.2	M	

Appendix B. Change History

(Informative)

B.1 Approved Version History

Reference	Date	Description
n/a	n/a	No previous version within OMA

B.2 Draft/Candidate Version 1.0 History

Document Identifier	Date	Sections	Description
Draft Versions OMA-WAP-OCSP-V1_0	17 Sep 2003	n/a	
	12 Jan 2004	Scope, 4, 8.2, 5.2, 5.3, 5.4.1, 5.4.2, 5.4.3, 5.5.2, 6.1.1, 6.1.3, 6.1.4, 6.3, App A, references,	Incorporates changes suggested by OMA REL & removal of Secure Time CR.
Candidate Version OMA-WAP-OCSP-V1_0	27 Jan 2004	n/a	Status changed to Candidate by TP TP ref # OMA-TP-2004-0020-OSCP-V1_0-for-candidate-approval

Appendix C. Example of OCSP messages

This section is informative.

This appendix contains examples of OCSP messages conforming to the profile specified herein. The messages are presented here in the value notation defined in [ISO/IEC 8824-1], a hexadecimal representation of their DER-encoding and an ASN.1 dump from the tool dumpasn1.

An example OCSP request:

```
exampleOCSPRequest OCSPRequest ::= {
  tbsRequest {
    version v1,
    requestList {
      {
        reqCert {
          hashAlgorithm {
            algorithm id-sha1,
            parameters SHA1Parameters : NULL
          },
          issuerNameHash '00E753C0C1BD92A40737E444D564790286942030'H,
          issuerKeyHash '165B8984F9355778C1222D38E3BE4C3DA4797B80'H,
          serialNumber 7470250387247641375
        }
      }
    },
    requestExtensions {
      {
        extnId id-pkix-ocsp-nonce,
        extnValue '313233343536373839303132333435363738393A'H
      }
    }
  }
}
```

The DER-encoded request becomes:

```
30 76 30 74 30 4B 30 49 30 47 30 07 06 05 2B 0E
03 02 1A 04 14 00 E7 53 C0 C1 BD 92 A4 07 37 E4
44 D5 64 79 02 86 94 20 30 04 14 16 5B 89 84 F9
35 57 78 C1 22 2D 38 E3 BE 4C 3D A4 79 7B 80 02
10 67 AB A9 25 EB 1E B3 1F C5 76 43 91 44 EC 41
19 A2 25 30 23 30 21 06 09 2B 06 01 05 05 07 30
01 02 04 14 31 32 33 34 35 36 37 38 39 30 31 32
33 34 35 36 37 38 39 30
```

The ASN.1 dump:


```

0 30 118: SEQUENCE {
  2 30 116: SEQUENCE {
    4 30 75: SEQUENCE {
      6 30 73: SEQUENCE {
        8 30 71: SEQUENCE {
          10 30 7: SEQUENCE {
            12 06 5: OBJECT IDENTIFIER sha1 (1 3 14 3 2 26)
              : }
          19 04 20: OCTET STRING
              : 00 E7 53 C0 C1 BD 92 A4 07 37 E4 44 D5 64 79 02
              : 86 94 20 30
          41 04 20: OCTET STRING
              : 16 5B 89 84 F9 35 57 78 C1 22 2D 38 E3 BE 4C 3D
              : A4 79 7B 80
          63 02 16: INTEGER
              : 67 AB A9 25 EB 1E B3 1F C5 76 43 91 44 EC 41 19
              : }
              : }
              : }
          81 A2 37: [2] {
            83 30 35: SEQUENCE {
              85 30 33: SEQUENCE {
                87 06 9: OBJECT IDENTIFIER '1 3 6 1 5 5 7 48 1 2'
                98 04 20: OCTET STRING
                    : 31 32 33 34 35 36 37 38 39 30 31 32 33 34 35 36
                    : 37 38 39 30
                    : }
                    : }
                    : }
                    : }
                    : }

```

An example OCSP response:

```

exampleBasicOCSPResponse BasicOCSPResponse ::= {
  tbsResponseData {
    version v1,
    responderID byName : rdnSequence : {
      {

```

```
{
  type id-at-countryName,
  value PrintableString : "SE"
}
},
{
  {
    type id-at-organizationName,
    value PrintableString : "OCSP-r-us.com"
  }
}
},
producedAt "20021202210000Z",
responses {
  {
    certID {
      hashAlgorithm {
        algorithm id-sha1,
        parameters SHA1Parameters : NULL
      },
      issuerNameHash '00E753C0C1BD92A40737E444D564790286942030'H,
      issuerKeyHash '165B8984F9355778C1222D38E3BE4C3DA4797B80'H,
      serialNumber 7470250387247641375
    },
    certStatus good : NULL,
    thisUpdate "20021202120000Z",
    nextUpdate "20021202180000Z"
  }
},
responseExtensions {
  {
    extnId id-pkix-ocsp-nonce,
    extnValue '313233343536373839303132333435363738393A'H
  }
}
},
signatureAlgorithm {
  algorithm sha1WithRSAEncryption,
  parameters SHA1WithRSAParameters : NULL
},
}
```

signature "H

}

The DER-encoded response becomes:

```

30 82 04 BF 0A 01 00 A0 82 04 B8 30 82 04 B4 06
09 2B 06 01 05 05 07 30 01 01 04 82 04 A5 30 82
04 A1 30 82 01 1B A1 6E 30 6C 31 17 30 15 06 03
55 04 0A 13 0E 56 65 72 69 53 69 67 6E 2C 20 49
6E 63 2E 31 12 30 10 06 03 55 04 0B 13 09 4F 43
53 50 20 54 65 73 74 31 1F 30 1D 06 03 55 04 0B
13 16 46 6F 72 20 54 65 73 74 20 50 75 72 70 6F
73 65 73 20 4F 6E 6C 79 31 1C 30 1A 06 03 55 04
03 13 13 4F 43 53 50 20 52 65 73 70 6F 6E 64 65
72 20 54 65 73 74 18 0F 32 30 30 30 30 39 30 31
31 37 33 31 34 38 5A 30 71 30 6F 30 47 30 07 06
05 2B 0E 03 02 1A 04 14 00 E7 53 C0 C1 BD 92 A4
07 37 E4 44 D5 64 79 02 86 94 20 30 04 14 16 5B
89 84 F9 35 57 78 C1 22 2D 38 E3 BE 4C 3D A4 79
7B 80 02 10 67 AB A9 25 EB 1E B3 1F C5 76 43 91
44 EC 41 19 80 00 18 0F 32 30 30 30 30 39 30 31
31 37 33 31 34 38 5A A0 11 18 0F 32 30 30 30 30
39 30 31 31 37 33 31 34 38 5A A1 25 30 23 30 21
06 09 2B 06 01 05 05 07 30 01 02 04 14 31 32 33
34 35 36 37 38 39 30 31 32 33 34 35 36 37 38 39
30 30 0B 06 09 2A 86 48 86 F7 0D 01 01 05 03 81
81 00 C8 DF 51 0E E0 E6 75 52 4F BC 1D 26 9D 2D
80 D6 5B 03 3C 33 E5 00 FB D5 57 00 9A 40 D3 2D
5C DB 6A 33 A8 7A 32 06 2F 23 14 B9 38 A3 50 8B
5C 96 63 4C B3 AD 31 FE 44 E5 7A B0 2F 15 E0 69
FC 10 F8 AE 77 07 AD E1 25 E6 32 59 21 5C 0F 34
91 6F 28 00 DB 8E A8 7E D8 C6 25 9D D0 B5 6F 2F
EC 4C 6E E5 39 E4 53 5A 62 E0 FC 62 C0 9C 24 4F
91 31 9C D2 9A BD AF F4 DB 46 AF 52 ED 67 D4 A7
0A 9E A0 82 02 ED 30 82 02 E9 30 82 02 E5 30 82
02 4E A0 03 02 01 02 02 11 00 FB FB 2A B2 1F 14
D4 FA EF D0 A9 ED 14 26 9E F5 30 0D 06 09 2A 86
48 86 F7 0D 01 01 05 05 00 30 4E 31 17 30 15 06
03 55 04 0A 13 0E 56 65 72 69 53 69 67 6E 2C 20
49 6E 63 2E 31 1F 30 1D 06 03 55 04 0B 13 16 46
6F 72 20 54 65 73 74 20 50 75 72 70 6F 73 65 73
20 4F 6E 6C 79 31 12 30 10 06 03 55 04 0B 13 09
4F 43 53 50 20 54 65 73 74 30 1E 17 0D 30 30 30
33 31 34 30 30 30 30 30 30 5A 17 0D 30 34 30 33
31 33 32 33 35 39 35 39 5A 30 6C 31 17 30 15 06
03 55 04 0A 13 0E 56 65 72 69 53 69 67 6E 2C 20
49 6E 63 2E 31 12 30 10 06 03 55 04 0B 13 09 4F
43 53 50 20 54 65 73 74 31 1F 30 1D 06 03 55 04
0B 13 16 46 6F 72 20 54 65 73 74 20 50 75 72 70
6F 73 65 73 20 4F 6E 6C 79 31 1C 30 1A 06 03 55
04 03 13 13 4F 43 53 50 20 52 65 73 70 6F 6E 64
65 72 20 54 65 73 74 30 81 9F 30 0D 06 09 2A 86
48 86 F7 0D 01 01 01 05 00 03 81 8D 00 30 81 89
02 81 81 00 D6 F7 64 A6 22 89 1E 3C 45 71 5F CB
58 EE A0 3F 5C 78 07 10 B0 79 4B 8A 41 5D 3E D4
6A 04 50 48 85 B0 28 81 31 69 79 DD D9 C8 86 67
76 7F 98 08 23 BB 0D F2 00 23 27 5B 18 B3 F4 2D
0A 39 97 D1 57 4A 83 86 C2 CF EB B9 9B E1 90 C2
91 2E 50 AC B7 7F BD D4 DC 3D 1F 0D DB D1 AE 5D

```

```

2F 7E F4 5D EC 5C 26 B8 A5 39 F7 20 81 8F F9 CB
DB D0 93 78 40 19 85 14 97 E6 EA CB FA CF 94 45
B6 5A 9C 4F 02 03 01 00 01 A3 81 A4 30 81 A1 30
24 06 03 55 1D 11 04 1D 30 1B A4 19 30 17 31 15
30 13 06 03 55 04 03 13 0C 50 69 6C 6F 74 4F 43
53 50 31 2D 31 30 4C 06 03 55 1D 1F 04 45 30 43
30 41 A0 3F A0 3D 86 3B 68 74 74 70 3A 2F 2F 6F
6E 73 69 74 65 63 72 6C 2E 76 65 72 69 73 69 67
6E 2E 63 6F 6D 2F 56 65 72 69 53 69 67 6E 49 6E
63 4F 43 53 50 54 65 73 74 2F 4C 61 74 65 73 74
43 52 4C 30 13 06 03 55 1D 25 04 0C 30 0A 06 08
2B 06 01 05 05 07 03 09 30 09 06 03 55 1D 13 04
02 30 00 30 0B 06 03 55 1D 0F 04 04 03 02 06 C0
30 0D 06 09 2A 86 48 86 F7 0D 01 01 05 05 00 03
81 81 00 B8 DB AE 66 41 71 7A B2 DE 71 4C E9 2C
F9 9F 76 B8 56 A9 9E 57 8B AD 39 0E 89 64 56 10
B1 FA DE 6F CC BC 7E 91 6E B0 43 B6 A0 40 D8 2E
76 B0 E9 A1 76 9E FB 80 07 F2 CE 84 64 06 77 74
14 DF A4 51 B4 86 86 89 B0 D6 37 6E D2 0B 67 C6
9A D9 A3 EC E7 B5 D0 2E 27 DE E6 06 78 65 77 AD
1C 50 6C A8 E7 50 1D 41 85 23 A2 17 31 4F B2 D9
F8 B7 FE 1C B4 A3 DC B8 21 24 BC 0F 35 D1 5E F5
0A E2 64

```

The ASN.1 dump:

```

0 30 1215: SEQUENCE {
  4 0A 1:  ENUMERATED CRYPT_MODE_NONE (0)
  7 A0 1208:  [0] {
11 30 1204:  SEQUENCE {
15 06 9:  OBJECT IDENTIFIER '1 3 6 1 5 5 7 48 1 1'
26 04 1189:  OCTET STRING, encapsulates {
30 30 1185:  SEQUENCE {
34 30 283:  SEQUENCE {
38 A1 110:  [1] {
40 30 108:  SEQUENCE {
42 31 23:  SET {
44 30 21:  SEQUENCE {
46 06 3:  OBJECT IDENTIFIER
:  organizationName (2 5 4 10)
51 13 14:  PrintableString 'ABC, Inc.'
:  }
:  }
67 31 18:  SET {
69 30 16:  SEQUENCE {
71 06 3:  OBJECT IDENTIFIER
:  organizationalUnitName (2 5 4 11)
76 13 9:  PrintableString 'OCSP Test'
:  }
:  }
87 31 31:  SET {
89 30 29:  SEQUENCE {
91 06 3:  OBJECT IDENTIFIER
:  organizationalUnitName (2 5 4 11)
96 13 22:  PrintableString 'For Test Purposes Only'
:  }
}

```

```

:
120 31 28: SET {
122 30 26: SEQUENCE {
124 06 3: OBJECT IDENTIFIER commonName (2 5 4 3)
129 13 19: PrintableString 'OCSP Responder Test'
:
: }
:
: }
:
150 18 15: GeneralizedTime '20000901173148Z'
167 30 113: SEQUENCE {
169 30 111: SEQUENCE {
171 30 71: SEQUENCE {
173 30 7: SEQUENCE {
175 06 5: OBJECT IDENTIFIER sha1 (1 3 14 3 2 26)
:
: }
182 04 20: OCTET STRING
:
: 00 E7 53 C0 C1 BD 92 A4 07 37 E4 44 D5 64 79 02
:
: 86 94 20 30
204 04 20: OCTET STRING
:
: 16 5B 89 84 F9 35 57 78 C1 22 2D 38 E3 BE 4C 3D
:
: A4 79 7B 80
226 02 16: INTEGER
:
: 67 AB A9 25 EB 1E B3 1F C5 76 43 91 44 EC 41 19
:
: }
244 80 0: [0]
:
: Error: Object has zero length.
246 18 15: GeneralizedTime '20000901173148Z'
263 A0 17: [0] {
265 18 15: GeneralizedTime '20000901173148Z'
:
: }
:
: }
282 A1 37: [1] {
284 30 35: SEQUENCE {
286 30 33: SEQUENCE {
288 06 9: OBJECT IDENTIFIER '1 3 6 1 5 5 7 48 1 2'
299 04 20: OCTET STRING
:
: 31 32 33 34 35 36 37 38 39 30 31 32 33 34 35 36
:
: 37 38 39 30
:
: }
:
: }
:
: }
321 30 11: SEQUENCE {
323 06 9: OBJECT IDENTIFIER
:
: sha1withRSAEncryption (1 2 840 113549 1 1 5)
:
: }
334 03 129: BIT STRING 0 unused bits
:
: C8 DF 51 0E E0 E6 75 52 4F BC 1D 26 9D 2D 80 D6
:
: 5B 03 3C 33 E5 00 FB D5 57 00 9A 40 D3 2D 5C DB
:
: 6A 33 A8 7A 32 06 2F 23 14 B9 38 A3 50 8B 5C 96
:
: 63 4C B3 AD 31 FE 44 E5 7A B0 2F 15 E0 69 FC 10
:
: F8 AE 77 07 AD E1 25 E6 32 59 21 5C 0F 34 91 6F
:
: 28 00 DB 8E A8 7E D8 C6 25 9D D0 B5 6F 2F EC 4C
:
: 6E E5 39 E4 53 5A 62 E0 FC 62 C0 9C 24 4F 91 31
:
: 9C D2 9A BD AF F4 DB 46 AF 52 ED 67 D4 A7 0A 9E
466 A0 749: [0] {

```

```

470 30 745: SEQUENCE {
474 30 741:     SEQUENCE {
478 30 590:         SEQUENCE {
482 A0 3:             [0] {
484 02 1:                 INTEGER 2
:                     }
487 02 17:             INTEGER
:                 00 FB FB 2A B2 1F 14 D4 FA EF D0 A9 ED 14 26 9E
:                 F5
506 30 13:         SEQUENCE {
508 06 9:             OBJECT IDENTIFIER
:                 sha1withRSAEncryption (1 2 840 113549 1 1 5)
519 05 0:             NULL
:         }
521 30 78: SEQUENCE {
523 31 23:     SET {
525 30 21:         SEQUENCE {
527 06 3:             OBJECT IDENTIFIER
:                 organizationName (2 5 4 10)
532 13 14:             PrintableString 'ABC, Inc.'
:         }
:     }
548 31 31: SET {
550 30 29:     SEQUENCE {
552 06 3:         OBJECT IDENTIFIER
:             organizationalUnitName (2 5 4 11)
557 13 22:         PrintableString 'For Test Purposes Only'
:     }
: }
581 31 18: SET {
583 30 16:     SEQUENCE {
585 06 3:         OBJECT IDENTIFIER
:             organizationalUnitName (2 5 4 11)
590 13 9:         PrintableString 'OCSP Test'
:     }
: }
601 30 30: SEQUENCE {
603 17 13:     UTCTime '000314000000Z'
618 17 13:     UTCTime '040313235959Z'
: }
633 30 108: SEQUENCE {
635 31 23:     SET {
637 30 21:         SEQUENCE {
639 06 3:             OBJECT IDENTIFIER
:                 organizationName (2 5 4 10)
644 13 14:             PrintableString 'ABC, Inc.'
:         }
:     }
660 31 18: SET {
662 30 16:     SEQUENCE {
664 06 3:         OBJECT IDENTIFIER
:             organizationalUnitName (2 5 4 11)
669 13 9:         PrintableString 'OCSP Test'
:     }
: }
680 31 31: SET {
682 30 29:     SEQUENCE {

```

```

684 06 3:          OBJECT IDENTIFIER
          :          organizationalUnitName (2 5 4 11)
689 13 22:         PrintableString 'For Test Purposes Only'
          :          }
          :          }
713 31 28:         SET {
715 30 26:         SEQUENCE {
717 06 3:          OBJECT IDENTIFIER commonName (2 5 4 3)
722 13 19:         PrintableString 'OCSP Responder Test'
          :          }
          :          }
          :          }
743 30 159:        SEQUENCE {
746 30 13:         SEQUENCE {
748 06 9:          OBJECT IDENTIFIER
          :          rsaEncryption (1 2 840 113549 1 1 1)
759 05 0:          NULL
          :          }
761 03 141:        BIT STRING 0 unused bits, encapsulates {
765 30 137:        SEQUENCE {
768 02 129:        INTEGER
          :          00 D6 F7 64 A6 22 89 1E 3C 45 71 5F CB 58 EE A0
          :          3F 5C 78 07 10 B0 79 4B 8A 41 5D 3E D4 6A 04 50
          :          48 85 B0 28 81 31 69 79 DD D9 C8 86 67 76 7F 98
          :          08 23 BB 0D F2 00 23 27 5B 18 B3 F4 2D 0A 39 97
          :          D1 57 4A 83 86 C2 CF EB B9 9B E1 90 C2 91 2E 50
          :          AC B7 7F BD D4 DC 3D 1F 0D DB D1 AE 5D 2F 7E F4
          :          5D EC 5C 26 B8 A5 39 F7 20 81 8F F9 CB DB D0 93
          :          78 40 19 85 14 97 E6 EA CB FA CF 94 45 B6 5A 9C
          :          [ Another 1 bytes skipped ]
900 02 3:          INTEGER 65537
          :          }
          :          }
          :          }
905 A3 164:        [3] {
908 30 161:        SEQUENCE {
911 30 36:         SEQUENCE {
913 06 3:          OBJECT IDENTIFIER
          :          subjectAltName (2 5 29 17)
918 04 29:         OCTET STRING, encapsulates {
920 30 27:         SEQUENCE {
922 A4 25:         [4] {
924 30 23:         SEQUENCE {
926 31 21:         SET {
928 30 19:         SEQUENCE {
930 06 3:          OBJECT IDENTIFIER
          :          commonName (2 5 4 3)
935 13 12:         PrintableString 'PilotOCSP1-
1'
          :          }
          :          }
          :          }
          :          }
          :          }
          :          }
949 30 76:        SEQUENCE {
951 06 3:          OBJECT IDENTIFIER

```

```

:           cRLDistributionPoints (2 5 29 31)
956 04 69:           OCTET STRING, encapsulates {
958 30 67:             SEQUENCE {
960 30 65:               SEQUENCE {
962 A0 63:                 [0] {
964 A0 61:                   [0] {
966 86 59:                     [6]
:           'http://onsitecrl.ABC.com/ABCIncOCSPTes'
:           't/LatestCRL'
:           }
:         }
:       }
:     }
1027 30 19: SEQUENCE {
1029 06 3:   OBJECT IDENTIFIER
:         extKeyUsage (2 5 29 37)
1034 04 12: OCTET STRING, encapsulates {
1036 30 10:   SEQUENCE {
1038 06 8:     OBJECT IDENTIFIER '1 3 6 1 5 5 7 3 9'
:   }
: }
1048 30 9: SEQUENCE {
1050 06 3:   OBJECT IDENTIFIER
:         basicConstraints (2 5 29 19)
1055 04 2:   OCTET STRING, encapsulates {
1057 30 0:     SEQUENCE {}
:   }
1059 30 11: SEQUENCE {
1061 06 3:   OBJECT IDENTIFIER keyUsage (2 5 29 15)
1066 04 4:   OCTET STRING, encapsulates {
1068 03 2:     BIT STRING 6 unused bits
:         '11'B
:   }
: }
: }
1072 30 13: SEQUENCE {
1074 06 9:   OBJECT IDENTIFIER
:         sha1withRSAEncryption (1 2 840 113549 1 1 5)
1085 05 0:   NULL
: }
1087 03 129: BIT STRING 0 unused bits
: B8 DB AE 66 41 71 7A B2 DE 71 4C E9 2C F9 9F 76
: B8 56 A9 9E 57 8B AD 39 0E 89 64 56 10 B1 FA DE
: 6F CC BC 7E 91 6E B0 43 B6 A0 40 D8 2E 76 B0 E9
: A1 76 9E FB 80 07 F2 CE 84 64 06 77 74 14 DF A4
: 51 B4 86 86 89 B0 D6 37 6E D2 0B 67 C6 9A D9 A3
: EC E7 B5 D0 2E 27 DE E6 06 78 65 77 AD 1C 50 6C
: A8 E7 50 1D 41 85 23 A2 17 31 4F B2 D9 F8 B7 FE
: 1C B4 A3 DC B8 21 24 BC 0F 35 D1 5E F5 0A E2 64
: }
: }
: }

```


