Enabler Test Specification for Online Certificate Status Protocol (OCSP) Mobile Profile

Candidate Version 1.0 – 20 Oct 2006

Open Mobile Alliance

OMA-ETS-OCSP_Mobile_Profile-V1_0-20061020-C

Contents

1. Scope

This document describes in detail available test cases for OCSP-MP V1.0.

The test cases are split in two categories, conformance and interoperability test cases.

The conformance test cases are aimed to verify the adherence to normative requirements described in the technical specifications.

The interoperability test cases are aimed to verify that implementations of the specifications work satisfactory.

If either conformance or interoperability tests do not exists at the creation of the test specification this part should be marked not available.

2. References

2.1 Normative References

[IOPPROC] "OMA Interoperability Policy and Process", Version 1.3, Open Mobile Alliance™, OMA-IOP-Process-V1_3, http://www.openmobilealliance.org/

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

[CERT] "WAP Certificate Profile", WAP-211-WAPCert, http://www.openmobilealliance.org/

[ERELD] "Enabler Relese Document for OCSP", Open Mobile Alliance™, OMA-ERELD-OCSP-V1_0, http://www.openmobilealliance.org/

[OCSPMP] "Online Certificate Status Protocol Mobile Profile", Open Mobile Alliance™, OMA-WAP-OCSP-V1_0, http://www.openmobilealliance.org/

2.2 Informative References

n/a

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”, are normative, unless they are explicitly indicated to be informative.

The following numbering scheme is used:

OCSP-1.0-con-number where:

y.z	Version of enabler release, e.g. 1.2 or 1.2.1
’con’	Indicating this test is a conformance test case
number	Leap number for the test case

Or

OCSP-1.0-int-number where:

y.z	Version of enabler release, e.g. 1.2 or 1.2.1
’int’	Indicating this test is a interoperability test case
number	Leap number for the test case

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.
Responder	A synomym for **Server**

3.3 Abbreviations

OMA	Open Mobile Alliance
OCSP MP	Online Certificate Status Protocol Mobile Profile
CA	Certificate Authority

4. Introduction

The purpose of this document is to provide test cases for OCSP Enabler Release 1.0

Some features in the OCSP V1.0 enabler may optionally be implemented in mobile devices. The tests associated with these optional features are marked as [Optional] in the test specification.

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.

5. Interoperability Test Cases

5.1 Valid certificate

Test Case Id	OCSP-1.0-int-01
Test Object	Client/server
Test Case Description	Client generates request for a valid certificate. Client receives and processes a response with a "good" status
Specification Reference	[OCSPMP] Chapters 5.1, 5.2, 5.3 and sections 5.5.1, 5.5.2,
SCR Reference	OCSP-C-001, OCSP-C-002, OCSP-C-005, OCSP-C-006, 0CSP-C-007, OCSP-C-009, OCSP-C-011, OCSP-C-012, OCSP-C-029, OCSP-C-030, OCSP-C-031, OCSP-C-033, OCSP-C-035, OCSP-C-037 OCSP-C-022, OCSP-C-003b
Tool	None
Test code	None
Preconditions	• Equipment: o 1 client/1 server • State: o Client trusts test root and has access to "valid" certificate o Server configured with response signing key and associated certificate issued from test intermediate CA o Both client and server have network connectivity.
Test Procedure	1. Presented with the "valid" certificate, client determines URL of server based on OCSP AIA extension within it. 2. Client generates **OCSPRequest** and formats request into HTTP GET request, sends to server and waits for response 3. Upon receiving response, clients validates **OCSPResponse** and reports back a "good" response
Pass-Criteria	1. Client properly generates **OCSPRequest** a. Uses SHA1 Algorithm b. Does not truncate hash values c. Should not contain a **RequestorName** field (this is an optional requirement) d. Properly formats HTTP GET message (base64 and url-encoded) 2. Clients sends GET request to server specified in the OCSPAIA URL contained within the "valid" certificate. 3. Client properly processes returned **OCSPResponse** from server a. Parse/process incoming **OCSPResponse** b. Verify **sha1WithRSAEncryption** signature on response using certificate contained with **OCSPResponse** structure (delegation support) c. Reports a "good" status.

5.2 Revoked certificate

Test Case Id	OCSP-1.0-int-02
Test Object	Client /server
Test Case Description	Client generates request for a revoked certificate. Client receives and processes a response with a "revoked" status
Specification Reference	[OCSPMP] Chapters 5.1, 5.2, 5.3 and sections 5.5.1, 5.5.2and 5.4.2
SCR Reference	OCSP-C-001, OCSP-C-002, OCSP-C-005, OCSP-C-006, 0CSP-C-007, OCSP-C-009, OCSP-C-011, OCSP-C-012, OCSP-C-029, OCSP-C-030, OCSP-C-031, OCSP-C-033, OCSP-C-035, OCSP-C-037 OCSP-C-023
Tool	None
Test code	None
Preconditions	• Equipment: o 1 client/1 server • State: o Client trusts test root and has access to "revoked" certificate o Server configured with response signing key and associated certificate issued from test intermediate CA o Both client and server have network connectivity.
Test Procedure	1. Presented with the "revoked" certificate, client determines URL of server based on OCSP AIA extension within it. 2. Client generates **OCSPRequest** and formats request into HTTP GET request, sends to server and waits for response 3. Upon receiving response, clients validates **OCSPResponse** and reports back a "revoked" response
Pass-Criteria	1. Client properly generates **OCSPRequest** a. Uses SHA1 Algorithm b. Does not truncate hash values c. Properly formats HTTP GET message (base64 and url-encoded) 2. Clients sends GET request to server specified in the OCSP AIA URL contained within the "revoked" certificate. 3. Client properly processes returned **OCSPResponse** from server a. Parse/process incoming **OCSPResponse** b. Verify **sha1WithRSAEncryption** signature on response using certificate contained with **OCSPResponse** structure (delegation support) c. Reports a "revoked" status.

5.3 Unknown certificate

Test Case Id	OCSP-1.0-int-03

Test Object	Client /server
Test Case Description	Client generates a request for a certificate unknown by the responder. Client receives and processes a response with an "unknown" status.
Specification Reference	[OCSPMP] Chapters 5.1, 5.2, 5.3 and sections 5.5.1, 5.5.2and 5.4.2
SCR Reference	OCSP-C-001, OCSP-C-002, OCSP-C-005, OCSP-C-006, 0CSP-C-007, OCSP-C-009, OCSP-C-011, OCSP-C-012, OCSP-C-029, OCSP-C-030, OCSP-C-031, OCSP-C-033, OCSP-C-035, OCSP-C-037 OCSP-C-020, OCSP-C-024, OCSP-C024a
Tool	None
Test code	None
Preconditions	• Equipment: o 1 client/1 server • State: o Client trusts test root and has access to "unknown" certificate o Server configured with response signing key and associated certificate issued from test intermediate CA o Both client and server have network connectivity.
Test Procedure	1. Presented with the "unknown" certificate, client determines URL of server based on OCSP AIA extension within it. 2. Client generates **OCSPRequest** and formats request into HTTP GET request, sends to server and waits for response 3. Upon receiving response, clients validates **OCSPResponse** and reports back a "unknown" response
Pass-Criteria	1. Client properly generates **OCSPRequest** a. Uses SHA1 Algorithm b. Does not truncate hash values c. Properly formats HTTP GET message (base64 and url-encoded) 2. Clients sends GET request to server specified in the OCSP AIA URL contained within the "unknown" certificate. 3. Client properly processes returned **OCSPResponse** from server a. Parse/process incoming **OCSPResponse** b. Verify **sha1WithRSAEncryption** signature on response using certificate contained with **OCSPResponse** structure (delegation support) c. Reports a "unknown" status.

5.4 Valid certificate containing a nonce

Test Case Id	OCSP-1.0-int-04
Test Object	Client/server

Test Case Description	Client generates a request for a valid certificate that contains a nonce. Client receives and processes a response with a "good" status but that does not contain a nonce.
Specification Reference	[OCSPMP] Chapters 5.1, 5.2, 5.3 and sections 5.5.1, 5.5.2and 5.4.1
SCR Reference	OCSP-C-001, OCSP-C-002, OCSP-C-005, OCSP-C-006, 0CSP-C-007, OCSP-C-009, OCSP-C-011, OCSP-C-012, OCSP-C-029, OCSP-C-030, OCSP-C-031, OCSP-C-033, OCSP-C-035, OCSP-C-037 OCSP-C-021
Tool	None
Test code	None
Preconditions	• Equipment: o 1 client/1 server • State: o Client trusts test root and has access to "valid" certificate o Server configured with response signing key and associated certificate issued from test intermediate CA o Both client and server have network connectivity.
Test Procedure	1. Presented with the "valid" certificate, client determines URL of server based on OCSP AIA extension within it. 2. Client generates **OCSPRequest** that contains a nonce and formats request into HTTP GET request, sends to server and waits for response 3. Upon receiving response, clients validates the **OCSPResponse** that does not contain a nonce and after determining although the nonce is not present the response is still valid (fresh) based on the local time in the clock and the validity values in the response
Pass-Criteria	1. Client properly generates **OCSPRequest** a. Uses SHA1 Algorithm b. Does not truncate hash values c. Includes a randomly generated nonce in an nonce extension d. Properly formats HTTP GET message (base64 and url-encoded) 2. Clients sends GET request to server specified in the OCSPAIA URL contained within the "valid" certificate. 3. Client properly processes returned **OCSPResponse** from server a. Parse/process incoming **OCSPResponse** b. Verify **sha1WithRSAEncryption** signature on response using certificate contained with **OCSPResponse** structure (delegation support) c. Determines nonce is not present d. Checks that the response is still fresh based on the local time in the clock and the validity values in the response

5.5 Signed request for a valid certificate

Test Case Id	OCSP-1.0-int-05
Test Object	Client/server
Test Case Description	Client generates a signed request for a valid certificate. Client receives and processes a response with a "good" status
Specification Reference	[OCSPMP] Chapters 5.1, 5.2, 5.3 and sections 5.5.1 and 5.5.2
SCR Reference	OCSP-C-001, OCSP-C-002, OCSP-C-005, OCSP-C-006, 0CSP-C-007, OCSP-C-009, OCSP-C-011, OCSP-C-012, OCSP-C-029, OCSP-C-030, OCSP-C-031, OCSP-C-033, OCSP-C-035, OCSP-C-037 OCSP-C-003, OCSP-C-003a
Tool	None
Test code	None
Preconditions	• Equipment: o 1 client/1 server • State: o Client trusts test root and has access to "valid" certificate o Client has a key and cert it can use to sign request o Server configured with response signing key and associated certificate issued from test intermediate CA o Both client and server have network connectivity.
Test Procedure	1. Presented with the "valid" certificate, client determines URL of server based on OCSP AIA extension within it. 2. Client generates a signed **OCSPRequest** and formats request into HTTP GET request, sends to server and waits for response 3. Upon receiving response, clients validates **OCSPResponse** and reports back a "good" response
Pass-Criteria	1. Client properly generates **OCSPRequest** a. Uses SHA1 Algorithm b. Does not truncate hash values c. Signs OCSPRequset d. Properly formats HTTP GET message (base64 and url-encoded) 2. Clients sends GET request to server specified in the OCSPAIA URL contained within the "valid" certificate. 3. Client properly processes returned **OCSPResponse** from server a. Parse/process incoming **OCSPResponse** b. Verify **sha1WithRSAEncryption** signature on response using certificate contained with **OCSPResponse** structure (delegation support) c. Reports a "good" status.

5.6 Generation of a request for avalid certificate with a nonce

Test Case Id	OCSP-1.0-int-06
Test Object	Client/server
Test Case Description	Client generates a request for a valid certificate that contains a nonce. Client receives and processes a response with a "good" status also containing a nonce
Specification Reference	[OCSPMP] Chapters 5.1, 5.2, 5.3 and sections 5.5.1 and 5.5.2
SCR Reference	OCSP-C-001, OCSP-C-002, OCSP-C-005, OCSP-C-006, 0CSP-C-007, OCSP-C-009, OCSP-C-011, OCSP-C-012, OCSP-C-029, OCSP-C-030, OCSP-C-031, OCSP-C-033, OCSP-C-035, OCSP-C-037 OCSP-C-004, OCSP-C-013
Tool	None
Test code	None
Preconditions	• Equipment: o 1 client/1 server • State: o Client trusts test root and has access to "valid" certificate o Server configured with response signing key and associated certificate issued from test intermediate CA o Both client and server have network connectivity.
Test Procedure	4. Presented with the "valid" certificate, client determines URL of server based on OCSP AIA extension within it. 5. Client generates **OCSPRequest** that contains a nonce and formats request into HTTP GET request, sends to server and waits for response 6. Upon receiving response, clients validates **OCSPResponse** (including the nonce) and reports back a "good" response

Pass-Criteria	4. Client properly generates **OCSPRequest** a. Uses SHA1 Algorithm b. Does not truncate hash values c. Includes a randomly generated nonce in an nonce extension d. Properly formats HTTP GET message (base64 and url-encoded) 5. Clients sends GET request to server specified in the OCSPAIA URL contained within the "valid" certificate. 6. Client properly processes returned **OCSPResponse** from server a. Parse/process incoming **OCSPResponse** b. Verify **sha1WithRSAEncryption** signature on response using certificate contained with **OCSPResponse** structure (delegation support) c. Ensures that the nonce it placed in the request matches the nonce it received in the response. d. Reports a "good" status.

5.7 Request with base64 and url encoded

Test Case Id	OCSP-1.0-int-07
Test Object	Client/server
Test Case Description	Client generates an OCSPRequest message that, when base64 and url-encoded, has a length of over 255 characters. Client receives and processes a response with a "good" status.
Specification Reference	[OCSPMP] Chapters 5.1, 5.2, 5.3 and sections 5.5.1 and 5.5.2
SCR Reference	OCSP-C-001, OCSP-C-002, OCSP-C-005, OCSP-C-006, 0CSP-C-007, OCSP-C-009, OCSP-C-011, OCSP-C-012, OCSP-C-029, OCSP-C-030, OCSP-C-031, OCSP-C-033, OCSP-C-035, OCSP-C-037 OCSP-C-034
Tool	None
Test code	None
Preconditions	• Equipment: o 1 client/1 server • State: o Client trusts test root and has access to "valid" certificate o Server configured with response signing key and associated certificate issued from test intermediate CA o Both client and server have network connectivity.

Test Procedure	1. Presented with the "valid" certificate, client determines URL of server based on OCSP AIA extension within it. 2. Client generates **OCSPRequest** that is longer thatn 255 characters, formats request into HTTP POST request, sends to server and waits for response 3. Upon receiving response, clients validates **OCSPResponse** and reports back a "good" response
Pass-Criteria	1. Client properly generates a large (perhaps signed and with unknown but non-critical request extensions) **OCSPRequest** a. Uses SHA1 Algorithm b. Does not truncate hash values c. Properly formats HTTP POST message (base64 and url-encoded) 2. Clients sends POST request to server specified in the OCSPAIA URL contained within the "valid" certificate. 3. Client properly processes returned **OCSPResponse** from server a. Parse/process incoming **OCSPResponse** b. Verify **sha1WithRSAEncryption** signature on response using certificate contained with **OCSPResponse** structure (delegation support) c. Reports a "good" status.

5.8 Valid certificate via TLS

Test Case Id	OCSP-1.0-int-08
Test Object	Client/server
Test Case Description	Client generates request for a valid certificate and sends it via TLS. Client receives and processes a response with a "good" status.
Specification Reference	[OCSPMP] Chapters 5.1, 5.2, 5.3 and sections 5.5.1 and 5.5.2
SCR Reference	OCSP-C-001, OCSP-C-002, OCSP-C-005, OCSP-C-006, 0CSP-C-007, OCSP-C-009, OCSP-C-011, OCSP-C-012, OCSP-C-029, OCSP-C-030, OCSP-C-031, OCSP-C-033, OCSP-C-035, OCSP-C-037 OCSP-C-032
Tool	None
Test code	None
Preconditions	• Equipment: o 1 client/1 server • State: o Client trusts test root and has access to "valid" certificate o Server configured with response signing key and associated certificate issued from test intermediate CA and has the ability to accept an incoming TLS request. o Both client and server have network connectivity.

Test Procedure	1. Presented with the "valid" certificate, client determines URL of server based on OCSP AIA extension within it. 2. Client generates **OCSPRequest** and formats request into HTTP GET request, sends to server and waits for response 3. Upon receiving response, clients validates **OCSPResponse** and reports back a "good" response
Pass-Criteria	1. Client properly generates **OCSPRequest** a. Uses SHA1 Algorithm b. Does not truncate hash values c. Properly formats HTTP GET message (base64 and url-encoded) 2. Clients sends GET request over TLS to server specified in the OCSPAIA URL contained within the "valid" certificate. 3. Client properly processes returned **OCSPResponse** from server d. Parse/process incoming **OCSPResponse** e. Verify **sha1WithRSAEncryption** signature on response using certificate contained with **OCSPResponse** structure (delegation support) f. Reports a "good" status.

6. Conformance Test Cases

6.1 Valid certificate with unknown extentions

Test Case Id	OCSP-1.0-con-01
Test Object	Client/server
Test Case Description	Client generates request for a valid certificate. Client receives and processes a response with a "good" status that contains unknown but non-critical extensions in the response.
Specification Reference	[OCSPMP] Chapters 5.1, 5.2, 5.3 and sections 5.5.1, 5.5.2
SCR Reference	OCSP-C-001, OCSP-C-002, OCSP-C-005, OCSP-C-006, 0CSP-C-007, OCSP-C-009, OCSP-C-011, OCSP-C-012, OCSP-C-029, OCSP-C-030, OCSP-C-031, OCSP-C-033, OCSP-C-035, OCSP-C-037 OCSP-C-015, OCSP-C-016
Tool	None
Test code	None
Preconditions	• Equipment: o 1 client/1 server • State: o Client trusts test root and has access to "good" certificate o Server configured with response signing key and associated certificate issued from test intermediate CA o Server responds with an OCSP Response that contains enough non-critical unknown extension such that the size of the response if close to 3000bytes large. o Both client and server have network connectivity.
Test Procedure	1. Presented with the "good" certificate, client determines URL of server based on OCSP AIA extension within it. 2. Client generates **OCSPRequest** and formats request into HTTP GET request, sends to server and waits for response 3. Upon receiving response, clients validates **OCSPResponse** and reports back a "good" response

Pass-Criteria	1. Client properly generates **OCSPRequest** a. Uses SHA1 Algorithm b. Does not truncate hash values c. Properly formats HTTP GET message (base64 and url-encoded) 2. Clients sends GET request to server specified in the OCSP AIA URL contained within the "good" certificate. 3. Client properly processes returned **OCSPResponse** from server a. Parse/process incoming **OCSPResponse** b. Clients does not fail because of the large response and the existence of the non-critical and unknown extensions. c. Verify **sha1WithRSAEncryption** signature on response using certificate contained with **OCSPResponse** structure (delegation support) d. Reports a "good" status.

6.2 Valid certificate with expired status

Test Case Id	OCSP-1.0-con-02
Test Object	Client/server
Test Case Description	Client generates request for a valid certificate. Client receives and processes a response with a "good" status but that has expired
Specification Reference	[OCSPMP] Chapters 5.1, 5.2, 5.3 and sections 5.5.1, 5.5.2, 5.4.1 and 5.4.2
SCR Reference	OCSP-C-001, OCSP-C-002, OCSP-C-005, OCSP-C-006, 0CSP-C-007, OCSP-C-009, OCSP-C-011, OCSP-C-012, OCSP-C-029, OCSP-C-030, OCSP-C-031, OCSP-C-033, OCSP-C-035, OCSP-C-037 OCSP-C-017, OCSP-C-022a
Tool	None
Test code	None
Preconditions	• Equipment: o 1 client/1 server • State: o Client trusts test root and has access to "valid" certificate o Server configured with response signing key and associated certificate issued from test intermediate CA o Server responds with an OCSP Rresponse "good" OCSPResponse that has expired (i.e. no longer valid) o Both client and server have network connectivity.
Test Procedure	1. Presented with the "valid" certificate, client determines URL of server based on OCSP AIA extension within it. 2. Client generates **OCSPRequest** and formats request into HTTP GET request, sends to server and waits for response 3. Upon receiving response, clients validates **OCSPResponse** and reports back a "good" response

Pass-Criteria	1. Client properly generates **OCSPRequest** a. Uses SHA1 Algorithm b. Does not truncate hash values c. Properly formats HTTP GET message (base64 and url-encoded) 2. Clients sends GET request to server specified in the OCSPAIA URL contained within the "valid" certificate. 3. Client properly processes returned **OCSPResponse** from server a. Parse/process incoming **OCSPResponse** b. Verify **sha1WithRSAEncryption** signature on response using certificate contained with **OCSPResponse** structure (delegation support) c. Reports that the response has expired and can't be trusted.

6.3 Valid certificate but delayed response

Test Case Id	OCSP-1.0-con-03
Test Object	Client/server
Test Case Description	Client generates request for a valid certificate. Client does not receive a response in a timely manner and thus times out and retries after a documented period of time.
Specification Reference	[OCSPMP] Chapters 5.1, 5.2, 5.3 and sections 5.5.1, 5.5.2and 5.4.2
SCR Reference	OCSP-C-001, OCSP-C-002, OCSP-C-005, OCSP-C-006, 0CSP-C-007, OCSP-C-009, OCSP-C-011, OCSP-C-012, OCSP-C-029, OCSP-C-030, OCSP-C-031, OCSP-C-033, OCSP-C-035, OCSP-C-037 OCSP-C-025, OCSP-C-026, OCSP-C-027
Tool	[OCSPMP] Chapters 5.1, 5.2, 5.3 and sections 5.5.1, 5.5.2and 5.4.3
Test code	None
Preconditions	• Equipment: o 1 client/1 server • State: o Client trusts test root and has access to "valid" certificate o Server is not available on/to the network
Test Procedure	1. Presented with the "valid" certificate, client determines URL of server based on OCSP AIA extension within it. 2. Client generates **OCSPRequest** and formats request into HTTP GET request, sends to server and waits for response

Pass-Criteria	1. Client properly generates **OCSPRequest** a. Uses SHA1 Algorithm b. Does not truncate hash values c. Properly formats HTTP GET message (base64 and url-encoded) 2. Clients sends GET request to server specified in the OCSPAIA URL contained within the "valid" certificate. 3. Client does not receive a response from the server in a timely manner and resends the request to the server after a set amount of time.

6.4 Nonce mismatch

Test Case Id	OCSP-1.0-con-04
Test Object	Client/server
Test Case Description	Client generates a request for a valid certificate that contains a nonce. Client receives and processes a response with a "good" that contains a nonce that does not match the nonce in the request
Specification Reference	[OCSPMP] Chapters 5.1, 5.2, 5.3 and sections 5.5.1, 5.5.2 and 5.4.1
SCR Reference	OCSP-C-001, OCSP-C-002, OCSP-C-005, OCSP-C-006, 0CSP-C-007, OCSP-C-009, OCSP-C-011, OCSP-C-012, OCSP-C-029, OCSP-C-030, OCSP-C-031, OCSP-C-033, OCSP-C-035, OCSP-C-037 OCSP-C-019
Tool	None
Test code	None
Preconditions	• Equipment: o 1 client/1 server • State: o Client trusts test root and has access to "valid" certificate o Server configured with response signing key and associated certificate issued from test intermediate CA o Both client and server have network connectivity.
Test Procedure	7. Presented with the "valid" certificate, client determines URL of server based on OCSP AIA extension within it. 8. Client generates **OCSPRequest** that contains a nonce and formats request into HTTP GET request, sends to server and waits for response 9. Upon receiving response, clients validates **OCSPResponse** (including the nonce) and after determining that the nonce values do not match reports back that the response is not valid

Pass-Criteria	7. Client properly generates **OCSPRequest** a. Uses SHA1 Algorithm b. Does not truncate hash values c. Includes a randomly generated nonce in an nonce extension d. Properly formats HTTP GET message (base64 and url-encoded) 8. Clients sends GET request to server specified in the OCSPAIA URL contained within the "valid" certificate. 9. Client properly processes returned **OCSPResponse** from server a. Parse/process incoming **OCSPResponse** b. Verify **sha1WithRSAEncryption** signature on response using certificate contained with **OCSPResponse** structure (delegation support) c. Attempts to match the nonce values d. Because nonce values do not match, client reports that the response is not valid.

6.5 Valid certificate not signed with sha1WithRSAEncryption

Test Case Id	OCSP-1.0-con-05
Test Object	Client/server
Test Case Description	Client generates request for a valid certificate. Client receives and processes a response with a "good" status signed with a signature algorithm other than **sha1WithRSAEncryption**
Specification Reference	[OCSPMP] Chapters 5.1, 5.2, 5.3 and sections 5.5.1 and 5.5.2
SCR Reference	OCSP-C-001, OCSP-C-002, OCSP-C-005, OCSP-C-006, 0CSP-C-007, OCSP-C-009, OCSP-C-011, OCSP-C-012, OCSP –C-028, OCSP-C-029, OCSP-C-030, OCSP-C-031, OCSP-C-033, OCSP-C-035, OCSP-C-037 OCSP-C-11a
Tool	None
Test code	None
Preconditions	• Equipment: o 1 client/1 server • State: o Client trusts test root and has access to "valid" certificate o Server configured with response signing key and associated certificate issued from test intermediate CA o Both client and server have network connectivity.

Test Procedure	1. Presented with the "valid" certificate, client determines URL of server based on OCSP AIA extension within it. 2. Client generates **OCSPRequest** and formats request into HTTP GET request, sends to server and waits for response 3. Upon receiving response signed with a signature algorithm other than **sha1WithRSAEncryption**, a clients validates **OCSPResponse** and reports back a "good" response.
Pass-Criteria	4. Client properly generates **OCSPRequest** a. Uses SHA1 Algorithm b. Does not truncate hash values c. Should not contain a **RequestorName** field (this is an optional requirement) d. Properly formats HTTP GET message (base64 and url-encoded) 5. Clients sends GET request to server specified in the OCSP AIA URL contained within the "valid" certificate. 6. Client properly processes returned **OCSPResponse** from server a. Parse/process incoming **OCSPResponse** b. Verify **md5WithRSAEncryption** signature on response using certificate contained with **OCSPResponse** structure (delegation support) c. Reports a "good" status.

Appendix A. Change History (Informative)

A.1 Approved Version History

Reference	Date	Description
n/a	n/a	No prior version –or- No previous version within OMA

A.2 Draft/Candidate Version 1.0 History

Document Identifier	Date	Sections	Description
Draft Versions OMA-ETS-OCSP_Mobile_Profile-V1_0	13 Jan 2005	All	First draft of this document.
	27 July 2005		Interoperability test cases completed
	09 Mar 2006	2.1, 5, 6	Incorporated CR: OMA-IOP-BRO-2006-0031R01
	04 Oct 2006	n/a	Submission for TP approval as OMA-TP-2006-0365R02-INP_OMA_ETS_OCSP_Mobile_Profile_V1_0_for_Approval_as_Candidate
Candidate Version OMA-ETS-OCSP_Mobile_Profile-V1_0	20 Oct 2006	n/a	Status changed to Candidate OMA-TP-2006-0365R02-INP_OMA_ETS_OCSP_Mobile_Profile_V1_0_for_Approval_as_Candidate (TP17)