KMIP Opaque Managed Object Store Profile Version 1.0

~~**Committee Specification Draft 01 /**~~
~~**Public Review Draft 01**~~

COMMENT CHANGES DRAFT

09 January 2014

Specification URIs

This version:

http://docs.oasis-open.org/kmip/kmip-opaque-obj-profile/v1.0/csprd01/kmip-opaque-obj-profile-v1.0-csprd01.doc (Authoritative)
http://docs.oasis-open.org/kmip/kmip-opaque-obj-profile/v1.0/csprd01/kmip-opaque-obj-profile-v1.0-csprd01.html
http://docs.oasis-open.org/kmip/kmip-opaque-obj-profile/v1.0/csprd01/kmip-opaque-obj-profile-v1.0-csprd01.pdf

Previous version:

N/A

Latest version:

~~http://docs.oasis-open.org/kmip/kmip-opaque-obj-profile/v1.0/kmip-opaque-obj-profile-v1.0.doc (Authoritative)~~
~~http://docs.oasis-open.org/kmip/kmip-opaque-obj-profile/v1.0/kmip-opaque-obj-profile-v1.0.html~~
~~http://docs.oasis-open.org/kmip/kmip-opaque-obj-profile/v1.0/kmip-opaque-obj-profile-v1.0.pdf~~
http://docs.oasis-open.org/kmip/kmip-opaque-obj-profile/v1.0/kmip-opaque-obj-profile-v1.0.doc (Authoritative)
http://docs.oasis-open.org/kmip/kmip-opaque-obj-profile/v1.0/kmip-opaque-obj-profile-v1.0.html
http://docs.oasis-open.org/kmip/kmip-opaque-obj-profile/v1.0/kmip-opaque-obj-profile-v1.0.pdf

Technical Committee:

OASIS Key Management Interoperability Protocol (KMIP) TC

Chairs:

Robert Griffin (robert.griffin@rsa.com), EMC Corporation
Subhash Sankuratripati (Subhash.Sankuratripati@netapp.com), NetApp

Editors:

Tim Hudson (tjh@cryptsoft.com), Cryptsoft Pty Ltd.
Robert Lockhart (Robert.Lockhart@thalesesec.com), Thales e-Security

Related work:

This specification is related to:

- *Key Management Interoperability Protocol Profiles Version 1.0*. 01 October 2010. OASIS Standard. http://docs.oasis-open.org/kmip/profiles/v1.0/os/kmip-profiles-1.0-os.html.
- *Key Management Interoperability Protocol Specification Version 1.1*. Latest version. http://docs.oasis-open.org/kmip/spec/v1.1/kmip-spec-v1.1.html.

- *Key Management Interoperability Protocol Specification Version 1.2.* Latest version. http://docs.oasis-open.org/kmip/spec/v1.2/kmip-spec-v1.2.html.

Abstract:

Describes a profile for a KMIP server performing opaque managed object storage operations based on requests received from a KMIP client.

Status:

This document was last revised or approved by the OASIS Key Management Interoperability Protocol (KMIP) TC on the above date. The level of approval is also listed above. Check the "Latest version" location noted above for possible later revisions of this document.

Technical Committee members should send comments on this specification to the Technical Committee's email list. Others should send comments to the Technical Committee by using the "Send A Comment" button on the Technical Committee's web page at ~~http://www.oasis-open.org/committees/kmip/~~http://www.oasis-open.org/committees/kmip/.

For information on whether any patents have been disclosed that may be essential to implementing this specification, and any offers of patent licensing terms, please refer to the Intellectual Property Rights section of the Technical Committee web page (~~http://www.oasis-open.org/committees/kmip/ipr.php~~http://www.oasis-open.org/committees/kmip/ipr.php).

Citation format:

When referencing this specification the following citation format should be used:

[kmip-opaque-obj-v1.0]

KMIP Opaque Managed Object Store Profile Version 1.0. Edited by Tim Hudson and Robert Lockhart. 09 January 2014. OASIS Committee Specification Draft 01 / Public Review Draft 01. http://docs.oasis-open.org/kmip/kmip-opaque-obj-profile/v1.0/csprd01/kmip-opaque-obj-profile-v1.0-csprd01.html. Latest version: ~~http://docs.oasis-open.org/kmip/kmip-opaque-obj-profile/v1.0/kmip-opaque-obj-profile-v1.0.html~~http://docs.oasis-open.org/kmip/kmip-opaque-obj-profile/v1.0/kmip-opaque-obj-profile-v1.0.html.

Notices

All capitalized terms in the following text have the meanings assigned to them in the OASIS Intellectual Property Rights Policy (the "OASIS IPR Policy"). The full Policy may be found at the OASIS website.

This document and translations of it may be copied and furnished to others, and derivative works that comment on or otherwise explain it or assist in its implementation may be prepared, copied, published, and distributed, in whole or in part, without restriction of any kind, provided that the above copyright notice and this section are included on all such copies and derivative works. However, this document itself may not be modified in any way, including by removing the copyright notice or references to OASIS, except as needed for the purpose of developing any document or deliverable produced by an OASIS Technical Committee (in which case the rules applicable to copyrights, as set forth in the OASIS IPR Policy, must be followed) or as required to translate it into languages other than English.

The limited permissions granted above are perpetual and will not be revoked by OASIS or its successors or assigns.

This document and the information contained herein is provided on an "AS IS" basis and OASIS DISCLAIMS ALL WARRANTIES, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO ANY WARRANTY THAT THE USE OF THE INFORMATION HEREIN WILL NOT INFRINGE ANY OWNERSHIP RIGHTS OR ANY IMPLIED WARRANTIES OF MERCHANTABILITY OR FITNESS FOR A PARTICULAR PURPOSE.

OASIS requests that any OASIS Party or any other party that believes it has patent claims that would necessarily be infringed by implementations of this OASIS Committee Specification or OASIS Standard, to notify OASIS TC Administrator and provide an indication of its willingness to grant patent licenses to such patent claims in a manner consistent with the IPR Mode of the OASIS Technical Committee that produced this specification.

OASIS invites any party to contact the OASIS TC Administrator if it is aware of a claim of ownership of any patent claims that would necessarily be infringed by implementations of this specification by a patent holder that is not willing to provide a license to such patent claims in a manner consistent with the IPR Mode of the OASIS Technical Committee that produced this specification. OASIS may include such claims on its website, but disclaims any obligation to do so.

OASIS takes no position regarding the validity or scope of any intellectual property or other rights that might be claimed to pertain to the implementation or use of the technology described in this document or the extent to which any license under such rights might or might not be available; neither does it represent that it has made any effort to identify any such rights. Information on OASIS' procedures with respect to rights in any document or deliverable produced by an OASIS Technical Committee can be found on the OASIS website. Copies of claims of rights made available for publication and any assurances of licenses to be made available, or the result of an attempt made to obtain a general license or permission for the use of such proprietary rights by implementers or users of this OASIS Committee Specification or OASIS Standard, can be obtained from the OASIS TC Administrator. OASIS makes no representation that any information or list of intellectual property rights will at any time be complete, or that any claims in such list are, in fact, Essential Claims.

The name "OASIS" is a trademark of OASIS, the owner and developer of this specification, and should be used only to refer to the organization and its official outputs. OASIS welcomes reference to, and implementation and use of, specifications, while reserving the right to enforce its marks against misleading uses. Please see http://www.oasis-open.org/policies-guidelines/trademark for above guidance.

Table of Contents

1 Introduction

For normative definition of the elements of KMIP see the KMIP Specification [KMIP-SPEC] and the KMIP Profiles [KMIP-PROF].

~~Illustrative guidance for the implementation of KMIP clients and servers is provided in the KMIP Usage Guide [KMIP-UG].~~

This profile defines the necessary KMIP functionality that a KMIP ~~server~~implementation conforming to this profile SHALL support in order to interoperate in conformance with this profile.

1.1 Terminology

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]**.

1.2 Normative References

[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.

~~**[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.~~

~~**[RFC2246]** T. Dierks and C. Allen, *The TLS Protocol, Version 1.0*, IETF RFC 2246, Jan 1999, http://www.ietf.org/rfc/rfc2246.txt~~**[KMIP-ENCODE]** *KMIP Additional Message Encodings Version 1.0.*
URL
Candidate OASIS Standard 01. DD MMM YYYY.

[KMIP-SPEC] One or more of [KMIP-SPEC-1_0], [KMIP-SPEC-1_1], [KMIP-SPEC-1_2]

[KMIP-SPEC-1_0] Key Management Interoperability Protocol Specification Version 1.0
http://docs.oasis-open.org/kmip/spec/v1.0/os/kmip-spec-1.0-os.doc
OASIS Standard, October 2010.

[KMIP-SPEC-1_1] *Key Management Interoperability Protocol Specification Version 1.1.*
http://docs.oasis-open.org/kmip/spec/v1.1/os/kmip-spec-v1.1-os.doc
OASIS Standard. 24 January 2013.

[KMIP-SPEC-1_2] *Key Management Interoperability Protocol Specification Version 1.2.*
URL
Candidate OASIS Standard 01. DD MMM YYYY.

[KMIP-PROF] One or more of [KMIP-PROF-1_0], [KMIP-PROF-1_1], [KMIP-PROF-1_2]

[KMIP-PROF-1_0] *Key Management Interoperability Protocol ~~Usage Guide~~Profiles Version 1.0*.
http://docs.oasis-open.org/kmip/profiles/v1.0/os/kmip-profiles-1.0-os.doc
OASIS Standard. 1 October 2010.

[KMIP-PROF-1_1] *Key Management Interoperability Protocol ~~Usage Guide~~Profiles Version 1.1*.
http://docs.oasis-open.org/kmip/profiles/v1.1/os/kmip-profiles-v1.1-os.doc
OASIS Standard 01. 24 January 2013.

[KMIP-PROF-1_2] *Key Management Interoperability Protocol ~~Usage Guide~~Profiles Version 1.2*.
URL
Candidate OASIS Standard 01. DD MMM YYYY.

~~1.3 Non-Normative References~~

~~**[KMIP-UG]** One or more of [KMIP-UG-1_0], [KMIP-UG-1_1], [KMIP-UG-1_2]~~

[KMIP-UG-1_0] *Key Management Interoperability Protocol Usage Guide Version 1.0.* http://docs.oasis-open.org/kmip/ug/v1.1/kmip-ug-v1.1-cnd01.doc Committee Note Draft, 1 December 2011

[KMIP-UG-1_1] *Key Management Interoperability Protocol Usage Guide Version 1.1.* http://docs.oasis-open.org/kmip/ug/v1.1/cn01/kmip-ug-v1.1-cn01.doc Committee Note 01, 27 July 2012

[KMIP-UG-1_2] *Key Management Interoperability Protocol Usage Guide Version 1.2.* URL Committee Note Draft, DD MMM YYYY

[KMIP-TC-1_1] *Key Management Interoperability Protocol Test Cases Version 1.1.* http://docs.oasis-open.org/kmip/testcases/v1.1/cn01/kmip-testcases-v1.1-cn01.doc, Committee Note 01, 27 July 2012.

[KMIP-TC-1_2] *Key Management Interoperability Protocol Test Cases Version 1.2.* URL, Committee Note Draft, DD MMM YYYY.

[KMIP-UC] *Key Management Interoperability Protocol Use Cases Version 1.0.* http://docs.oasis-open.org/kmip/usecases/v1.0/cs01/kmip-usecases-1.0-cs-01.doc, Committee Specification, 15 June 2010.

2 Opaque Managed Object Store Profile

The Opaque Managed Object Store Profile is a KMIP server performing storage related operations on opaque objects based on requests received from a KMIP client.

2.1 Authentication Suite

Implementations conformant to this profile SHALL support at least one of the Authentication Suites defined within section 3 of [KMIP-PROF. The establishment of the trust relationship between the KMIP client and the KMIP server is the same as the defined base profiles.

~~2.2 Baseline~~

2.2 Opaque Managed Object Store – Client

KMIP clients conformant to this profile~~:~~ under [KMIP-SPEC-1_0]:

1. SHALL conform to the [KMIP-SPEC-1_0]

KMIP clients conformant to this profile under [KMIP-SPEC-1_1]:

2. SHALL conform to the *Baseline Client Clause* (section 5.12) of [KMIP-PROF-1_1]

KMIP clients conformant to this profile under [KMIP-SPEC-1_2]:

~~1.~~3. SHALL conform to the Baseline Client ~~profile in~~(section 5.2) of [KMIP-PROF~~] and [KMIP-SPEC-1_2~~]

KMIP clients conformant to this profile:

4. MAY support any clause within [KMIP-SPEC] provided it does not conflict with any other clause within this section 1.1
5. MAY support extensions outside the scope of this standard (e.g., vendor extensions, conformance clauses) that do not contradict any KMIP requirements.

2.3 Opaque Managed Object Store – Server

KMIP servers conformant to this profile~~:~~ under [KMIP-SPEC-1_0]:

1. SHALL conform to the [KMIP-SPEC-1_0]

KMIP servers conformant to this profile under [KMIP-SPEC-1_1]:

~~1.~~2. SHALL conform to the *Baseline Server* ~~profile in~~of [KMIP-PROF~~] and [KMIP-SPEC] and~~1_1]

KMIP servers conformant to this profile under [KMIP-SPEC-1_2]:

3. SHALL conform to the Baseline Server of [KMIP-PROF-1_2]

KMIP servers conformant to this profile:

~~2.~~4. SHALL support the following *Objects* [KMIP-SPEC]
   a. *Opaque Object* [KMIP-SPEC]

~~3.~~5. SHALL support the following *Attributes* [KMIP-SPEC]
   a. Object Type [KMIP-SPEC]

~~4.~~6. SHALL support the following *Client-to-Server* [KMIP-SPEC] operations:
   a. *Register* [KMIP-SPEC]

~~5.~~7. SHALL support the following *Message Encoding* [KMIP-SPEC]:
   a. *Opaque Data Type* [KMIP-SPEC]
   b. *Object Type* [KMIP-SPEC] with value:

i. Opaque Object

~~6. SHALL support all Mandatory Test Cases, returning results in accordance with the test cases.~~

~~7.~~8. MAY support any clause within [KMIP-SPEC] provided it does not conflict with any other clause within this section ~~2.2~~2.3

~~8. MAY support extensions outside the scope of this standard (e.g., vendor extensions, conformance clauses) that do not contradict any KMIP requirements.~~

9. ~~Opaque Managed Object Store~~ MAY support extensions outside the scope of this standard (e.g., vendor extensions, conformance clauses) that do not contradict any KMIP requirements.

3 Opaque Managed Object Store Profile - Test Cases

~~This section documents the test cases for a KMIP server performing management and storage operations of opaque objects, based on requests received from a KMIP client.~~

The test cases define a number of request-response pairs for KMIP operations. Each test case is provided in the XML format specified in [KMIP-ENCODE] intended to be both human-readable and usable by automated tools. The time sequence (starting from 0) for each request-response pair is noted and line numbers are provided for ease of cross-reference for a given test sequence.

Each test case has a unique label (the section name) which includes indication of mandatory (-M-) or optional (-O-) status and the protocol version major and minor numbers as part of the identifier.

The test cases may depend on a specific configuration of a KMIP client and server being configured in a manner consistent with the test case assumptions.

Where possible the flow of unique identifiers between tests, the date-time values, and other dynamic items are indicated using symbolic identifiers – in actual request and response messages these dynamic values will be filled in with valid values.

Note: the values for the returned items and the custom attributes are illustrative. Actual values from a real client system ~~will vary.~~may vary as specified in section 4.7.

3.1 Mandatory Test Cases KMIP 1.0

~~This section documents the test cases that a client or server conformant to the Opaque Managed Object Store Profile SHALL support under KMIP Specification 1.0.~~

3.1.1 OMOS-M-1-10

Register small opaque object

```
      # TIME 0
0001  <RequestMessage>
0002    <RequestHeader>
0003      <ProtocolVersion>
0004        <ProtocolVersionMajor type="Integer" value="1"/>
0005        <ProtocolVersionMinor type="Integer" value="0"/>
0006      </ProtocolVersion>
0007      <BatchCount type="Integer" value="1"/>
0008    </RequestHeader>
0009    <BatchItem>
0010      <Operation type="Enumeration" value="Register"/>
0011      <RequestPayload>
0012        <ObjectType type="Enumeration" value="OpaqueObject"/>
0013        <TemplateAttribute>
0014          <Attribute>
0015            <AttributeName type="TextString" value="Name"/>
0016            <AttributeValue>
0017              <NameValue type="TextString" value="OMOS-M-1-10"/>
0018              <NameType type="Enumeration"
      value="UninterpretedTextString"/>
0019            </AttributeValue>
0020          </Attribute>
0021        </TemplateAttribute>
0022        <OpaqueObject>
0023          <OpaqueDataType type="Enumeration" value="0x80000001"/>
0024          <OpaqueDataValue type="ByteString"
      value="53656372657450617373776f7264"/>
```

```
0025        </OpaqueObject>
0026      </RequestPayload>
0027    </BatchItem>
0028  </RequestMessage>
0029  <ResponseMessage>
0030    <ResponseHeader>
0031      <ProtocolVersion>
0032        <ProtocolVersionMajor type="Integer" value="1"/>
0033        <ProtocolVersionMinor type="Integer" value="0"/>
0034      </ProtocolVersion>
0035      <TimeStamp type="DateTime" value="2012-04-27T08:12:24+00:00"/>
0036      <BatchCount type="Integer" value="1"/>
0037    </ResponseHeader>
0038    <BatchItem>
0039      <Operation type="Enumeration" value="Register"/>
0040      <ResultStatus type="Enumeration" value="Success"/>
0041      <ResponsePayload>
0042        <UniqueIdentifier type="TextString"
      value="$UNIQUE_IDENTIFIER_0"/>
0043      </ResponsePayload>
0044    </BatchItem>
0045  </ResponseMessage>
      # TIME 1
0046  <RequestMessage>
0047    <RequestHeader>
0048      <ProtocolVersion>
0049        <ProtocolVersionMajor type="Integer" value="1"/>
0050        <ProtocolVersionMinor type="Integer" value="0"/>
0051      </ProtocolVersion>
0052      <BatchCount type="Integer" value="1"/>
0053    </RequestHeader>
0054    <BatchItem>
0055      <Operation type="Enumeration" value="Destroy"/>
0056      <RequestPayload>
0057        <UniqueIdentifier type="TextString"
      value="$UNIQUE_IDENTIFIER_0"/>
0058      </RequestPayload>
0059    </BatchItem>
0060  </RequestMessage>
0061  <ResponseMessage>
0062    <ResponseHeader>
0063      <ProtocolVersion>
0064        <ProtocolVersionMajor type="Integer" value="1"/>
0065        <ProtocolVersionMinor type="Integer" value="0"/>
0066      </ProtocolVersion>
0067      <TimeStamp type="DateTime" value="2012-04-27T08:12:24+00:00"/>
0068      <BatchCount type="Integer" value="1"/>
0069    </ResponseHeader>
0070    <BatchItem>
0071      <Operation type="Enumeration" value="Destroy"/>
0072      <ResultStatus type="Enumeration" value="Success"/>
0073      <ResponsePayload>
0074        <UniqueIdentifier type="TextString"
      value="$UNIQUE_IDENTIFIER_0"/>
0075      </ResponsePayload>
0076    </BatchItem>
0077  </ResponseMessage>
```

3.2 Mandatory Test Cases KMIP 1.1

~~This section documents the test cases that a client or server conformant to the Opaque Managed Object Store Profile SHALL support under KMIP Specification 1.1.~~

3.2.1 OMOS-M-1-11

Register small opaque object

```
      # TIME 0
0001  <RequestMessage>
0002    <RequestHeader>
0003      <ProtocolVersion>
0004        <ProtocolVersionMajor type="Integer" value="1"/>
0005        <ProtocolVersionMinor type="Integer" value="1"/>
0006      </ProtocolVersion>
0007      <BatchCount type="Integer" value="1"/>
0008    </RequestHeader>
0009    <BatchItem>
0010      <Operation type="Enumeration" value="Register"/>
0011      <RequestPayload>
0012        <ObjectType type="Enumeration" value="OpaqueObject"/>
0013        <TemplateAttribute>
0014          <Attribute>
0015            <AttributeName type="TextString" value="Name"/>
0016            <AttributeValue>
0017              <NameValue type="TextString" value="OMOS-M-1-11"/>
0018              <NameType type="Enumeration"
      value="UninterpretedTextString"/>
0019            </AttributeValue>
0020          </Attribute>
0021        </TemplateAttribute>
0022        <OpaqueObject>
0023          <OpaqueDataType type="Enumeration" value="0x80000001"/>
0024          <OpaqueDataValue type="ByteString"
      value="53656372657450617373776f7264"/>
0025        </OpaqueObject>
0026      </RequestPayload>
0027    </BatchItem>
0028  </RequestMessage>
0029  <ResponseMessage>
0030    <ResponseHeader>
0031      <ProtocolVersion>
0032        <ProtocolVersionMajor type="Integer" value="1"/>
0033        <ProtocolVersionMinor type="Integer" value="1"/>
0034      </ProtocolVersion>
0035      <TimeStamp type="DateTime" value="2012-04-27T08:12:24+00:00"/>
0036      <BatchCount type="Integer" value="1"/>
0037    </ResponseHeader>
0038    <BatchItem>
0039      <Operation type="Enumeration" value="Register"/>
0040      <ResultStatus type="Enumeration" value="Success"/>
0041      <ResponsePayload>
0042        <UniqueIdentifier type="TextString"
      value="$UNIQUE_IDENTIFIER_0"/>
```

```
0043        </ResponsePayload>
0044      </BatchItem>
0045    </ResponseMessage>
```

```
        # TIME 1
0046    <RequestMessage>
0047      <RequestHeader>
0048        <ProtocolVersion>
0049          <ProtocolVersionMajor type="Integer" value="1"/>
0050          <ProtocolVersionMinor type="Integer" value="1"/>
0051        </ProtocolVersion>
0052        <BatchCount type="Integer" value="1"/>
0053      </RequestHeader>
0054      <BatchItem>
0055        <Operation type="Enumeration" value="Destroy"/>
0056        <RequestPayload>
0057          <UniqueIdentifier type="TextString"
        value="$UNIQUE_IDENTIFIER_0"/>
0058        </RequestPayload>
0059      </BatchItem>
0060    </RequestMessage>
```

```
0061    <ResponseMessage>
0062      <ResponseHeader>
0063        <ProtocolVersion>
0064          <ProtocolVersionMajor type="Integer" value="1"/>
0065          <ProtocolVersionMinor type="Integer" value="1"/>
0066        </ProtocolVersion>
0067        <TimeStamp type="DateTime" value="2012-04-27T08:12:24+00:00"/>
0068        <BatchCount type="Integer" value="1"/>
0069      </ResponseHeader>
0070      <BatchItem>
0071        <Operation type="Enumeration" value="Destroy"/>
0072        <ResultStatus type="Enumeration" value="Success"/>
0073        <ResponsePayload>
0074          <UniqueIdentifier type="TextString"
        value="$UNIQUE_IDENTIFIER_0"/>
0075        </ResponsePayload>
0076      </BatchItem>
0077    </ResponseMessage>
```

3.3 Mandatory Test Cases KMIP 1.2

~~This section documents the test cases that a client or server conformant to the Opaque Managed Object Store Profile SHALL support under KMIP Specification 1.2.~~

3.3.1 OMOS-M-1-12

Register small opaque object

```
        # TIME 0
0001    <RequestMessage>
0002      <RequestHeader>
0003        <ProtocolVersion>
0004          <ProtocolVersionMajor type="Integer" value="1"/>
0005          <ProtocolVersionMinor type="Integer" value="2"/>
0006        </ProtocolVersion>
0007        <BatchCount type="Integer" value="1"/>
```

```
0008    </RequestHeader>
0009    <BatchItem>
0010      <Operation type="Enumeration" value="Register"/>
0011      <RequestPayload>
0012        <ObjectType type="Enumeration" value="OpaqueObject"/>
0013        <TemplateAttribute>
0014          <Attribute>
0015            <AttributeName type="TextString" value="Name"/>
0016            <AttributeValue>
0017              <NameValue type="TextString" value="OMOS-M-1-12"/>
0018              <NameType type="Enumeration"
      value="UninterpretedTextString"/>
0019            </AttributeValue>
0020          </Attribute>
0021        </TemplateAttribute>
0022        <OpaqueObject>
0023          <OpaqueDataType type="Enumeration" value="0x80000001"/>
0024          <OpaqueDataValue type="ByteString"
      value="53656372657450617373776f7264"/>
0025        </OpaqueObject>
0026      </RequestPayload>
0027    </BatchItem>
0028  </RequestMessage>
0029  <ResponseMessage>
0030    <ResponseHeader>
0031      <ProtocolVersion>
0032        <ProtocolVersionMajor type="Integer" value="1"/>
0033        <ProtocolVersionMinor type="Integer" value="2"/>
0034      </ProtocolVersion>
0035      <TimeStamp type="DateTime" value="2012-04-27T08:12:24+00:00"/>
0036      <BatchCount type="Integer" value="1"/>
0037    </ResponseHeader>
0038    <BatchItem>
0039      <Operation type="Enumeration" value="Register"/>
0040      <ResultStatus type="Enumeration" value="Success"/>
0041      <ResponsePayload>
0042        <UniqueIdentifier type="TextString"
      value="$UNIQUE_IDENTIFIER_0"/>
0043      </ResponsePayload>
0044    </BatchItem>
0045  </ResponseMessage>
      # TIME 1
0046  <RequestMessage>
0047    <RequestHeader>
0048      <ProtocolVersion>
0049        <ProtocolVersionMajor type="Integer" value="1"/>
0050        <ProtocolVersionMinor type="Integer" value="2"/>
0051      </ProtocolVersion>
0052      <BatchCount type="Integer" value="1"/>
0053    </RequestHeader>
0054    <BatchItem>
0055      <Operation type="Enumeration" value="Destroy"/>
0056      <RequestPayload>
0057        <UniqueIdentifier type="TextString"
      value="$UNIQUE_IDENTIFIER_0"/>
0058      </RequestPayload>
0059    </BatchItem>
0060  </RequestMessage>
```

```
0061  <ResponseMessage>
0062    <ResponseHeader>
0063      <ProtocolVersion>
0064        <ProtocolVersionMajor type="Integer" value="1"/>
0065        <ProtocolVersionMinor type="Integer" value="2"/>
0066      </ProtocolVersion>
0067      <TimeStamp type="DateTime" value="2012-04-27T08:12:24+00:00"/>
0068      <BatchCount type="Integer" value="1"/>
0069    </ResponseHeader>
0070    <BatchItem>
0071      <Operation type="Enumeration" value="Destroy"/>
0072      <ResultStatus type="Enumeration" value="Success"/>
0073      <ResponsePayload>
0074        <UniqueIdentifier type="TextString"
      value="$UNIQUE_IDENTIFIER_0"/>
0075      </ResponsePayload>
0076    </BatchItem>
0077  </ResponseMessage>
```

144

145

146 3.4 Optional Test Cases KMIP 1.0

147 ~~This section documents the test cases that a client or server conformant to the Opaque Managed Object~~
148 ~~Store Profile MAY support under KMIP Specification 1.0.~~

149 3.4.1 OMOS-O-1-10

150 Register larger (>10k) opaque object

```
      # TIME 0
0001  <RequestMessage>
0002    <RequestHeader>
0003      <ProtocolVersion>
0004        <ProtocolVersionMajor type="Integer" value="1"/>
0005        <ProtocolVersionMinor type="Integer" value="0"/>
0006      </ProtocolVersion>
0007      <BatchCount type="Integer" value="1"/>
0008    </RequestHeader>
0009    <BatchItem>
0010      <Operation type="Enumeration" value="Register"/>
0011      <RequestPayload>
0012        <ObjectType type="Enumeration" value="OpaqueObject"/>
0013        <TemplateAttribute>
0014          <Attribute>
0015            <AttributeName type="TextString" value="Name"/>
0016            <AttributeValue>
0017              <NameValue type="TextString" value="OMOS-O-1-10"/>
0018              <NameType type="Enumeration"
      value="UninterpretedTextString"/>
0019            </AttributeValue>
0020          </Attribute>
0021        </TemplateAttribute>
0022        <OpaqueObject>
0023          <OpaqueDataType type="Enumeration" value="0x80000001"/>
0024          <OpaqueDataValue type="ByteString"
      value="168392816fd71b3d1c5d9cecfacf61f4e396374ede655d9d15305d6a0a04e
      5f0beab1de8be60fb716de00456c0b4adaadd5e1f4e72879251dbf7d25ca9f81076d
```

	aa0b6464ae989a76a6f6710ea9560a60b99cb4f697cd075cd799cb7dbcfffab4c2ab a5a19529f14307f6d217b1c84114eab50855b623d2e2a7602cdd230778939cce2a03 550b0e0c9a4ff7e0ad2af805a92bbe4a41ba3405565ca050c38c6d5b92d902c30544 b1460e2360459ee2ef3376b66caf91e0e0980d12ea6c19b5623cf03ad065652cf247 ee2be155deacfda3d96b35f21d2f97fe4fd28244dec67f61c32250f5fc93dc515c1b 5c7004f212b7c1d60972f3aa0372789364a3a762f80fda1d58389ea3cb3d204db887 b0db62623350d4ea7d1bfd91e6d522fab6942abe5ab9f76278e4cb280fed409268c2 731552c8292829a47355852d5780388a4e13691f8ec654226ce52e213fff30b0b3de 7ffb7444c7748f7e90dd893276d526a657bdf42ea588721788feb605e5d3443ebe06 91be98af902a3d6a459f1e160df7dc3a7507b05a238d49c6d5ef6803ffb964cd813d b90f549c2393fae94fcfc8c05ddb62a71bfc031074f4d32ada48491c970dedf57c13 9cb04c94112fcef3eec9fdd7487eecd1470741f780e0d9e99ba68e97945b7ab7970f 8003f80ca9622c94192281c13380894dc1f6c6d88848ffe81fd994862d2c60db1b65 1dbf12a245d34fc0e2a1b7cf36428c1e481890607a4df45ea20619ea02946af0c7c4 1fc16bc620159871659c8105506fb0d4017921ea79ac082afad5cb9bf703a49ac79f d1f428fdca5a8f693990bfbca9640a44bf43e5786111f624369d1d33a08cd7247be0 7f9bad26e531a2f3f9aea66536682296348cf86291af9c2521bc6196747986d02b3a 465dfbbe9468c3b364a8048441e32dc8ff190f0ad62b2c0f6a6d4aa715580b1fb2ca b038560259981eee6f59f850c076c29507e9efd9cca183f5bd8a0a876820ed173f88 4f6a9773c8102859b59b286002147b86428c18537590ff9b3cbe3dd2f7607f2b7d84 227c5d9ca6e6f272631672960f0da9c69d6b31eb499d50d724e3e3d4f0c982424389 15de54f1a600acd13f9ce483c01dcd8c5d36a7cc57ead5f4066b849b8a5e00cc5601 9bebb2e8129a6481e9f4b234154808236538c500ccd597a273068dc442f3c1205005 0209538cb1ea3194a18278b55d589d34e7680f565ed411359eaddf12c0bdfe907b16 377827d6ca46460d19f703c7b17fd3b7be9f45c162a34502f507673d304487e11c27 c9827cbc9cc5e3992514aa6e62d3163e338af19e99fcddec6ee17f70af7c9644a297 af3ec237619b8580bd20ae8ada4ce7dfe834711560b598f5aab65eac8fd2a8d66f57 4c79e4f19faa08dda5823ddc4f532f3053841dc52213dc147b325599a4b5969fa264 2fb50c01030b14253d0f23fd34f663581e95da9bfb0e3f52a010b2f5911ff063ac2a 826c94a8789445bc229a1ef1fe74fd8b5f672e8ffa671a5c69d19a4d7611c149cd52 68d590788aa3e44e949beb46f38a8fa51a301824e88c220ae41ba4b036c672342c04 3ebc6db91035b70ae68d58f558a3a5d1a788a6694c4f74278a204743fae6d947b502 e29552ba65507e91b684a3955aafb43530f02ccccaeded9a6d1f877da2470f36c260 14653493c970613f25de9c21f0682062881de0fc6ce2712e6f924408e7d29d368c43 ed198f14f5e947bb2f721084e6c22f750a6cd2400c49b9689e4e0f5d3d52005c5e42 a6aa0ecdc237f7b1868b7e77bce2c0d8f160a061f0de6529967e82586fe854bc89ed 68dc7d9ac521aa2f40e1c4c3835ad8c2881ec579975d4cf9d9beb69d0f2c4b4f1e69 ace6da5b9f6e49801ac9e1c5922176e3c72b8f0b73629d457a10456d9c8f0ae56e40 cb01f64ba0b49ea69f23728cffa532fd01b966ff31c30fedc9a52b7b0a5fe6a2e3bd 53c87560cf74b12143696e52343317c408d8e13bdb1fbd2758aaa3cf8aa3fd229bb6 5a9e41228372889147470cee02ae4acb89d52e31f8da66543fc8a6c02a6b337ce10d 0f41f464a44509c245476be1e5b11df8dbb867664cec0882beee6e21a022abcb89de d8a17cf6f17da33c094e42be30a3c070950ff300993abb40fac22f0ecf7d155c9261 516796cb1de4c0d6bfc21e716b04d23dbb8b465708204a3b96af8e46042e8205e7df 92b9e01c7794c94a5036b7f85fe0967515cc05ea460c7810c0c551d8fd94f7ad7d1a 641e73a1471e9364bf4e2b4687c7700381d46d39a3159f486d0323de59c5f555d323 e5c072fbfc0758e22aa6a1df04d13bf4bfe6632853424984d56d16ed61402a640c7e 1b07d9439d1fdc2df147df02639575b50da6cd769cb367dbfe3316e03939e85880f8 b19f2689d5504f25200560dd815fd4535a1b5f70c8332c95c9d292c75e971df28ec6 ff6d70a52ad78d236a51ec4a12243a650285fbd6aa4632fab8fa56eb26f638855147 d72efcdd8a0b4367122f8e2210d39ffc87ac29ab5c5f1226bea04693e0b5c671269d 96ac0c665fbd4f5fdb04ffcd7b76f0f8c1960e7a47c88226d6bd360e4dd65d70ed68 7e2fea04822f7c8394007ee085a9362a35506696c44e531786472a3db4bddd2b63ae 1a448d0442d11f28dfed4de000c820e40174e216d274ab1321be6f21d1e4eb7e2243 0c131b2050c1fcb9ebb2823dd6fb4f4972ea4167795f911fb8a3c7f14ecf71a0e675 657263c2f4b5eb72369d9d2d457999dc15392ccc10d98308830dd9b64d95427381af a9549c5df5081de88849126197154968d96797c573901e80bc638bc4126c01dea36a 56c1afe01021a21a776f6b4e375dd42156cce98998bd3036401550b501fac4ad653b

	cf098db8f6d9ef76429a60137e0f507c67b57ddf829c5ed88f4369a6cc7287683fa2 25515697cc2e43ffd8108e7f1564736d043b6323ae17bda3031fab7712f886ac12af beb2686700c1133017f64363293f93ab3c4e096aff76751377e5b6e5ead512a2f3d3 6635fcac28bf7fcf5c565bcf51bf650e7e3c80194a34b7348e7517fd4301f9986cdc bbe27392291da368ca699d90bca1b49122b20649f6b95529a72ba8217546042cb975 720b3c6e08ea2ed9faf23f975524a3857f5d20f76f6df0b5cf16fe7b054f4c996042 728c8b41326f7edf94fe04b0c084762ecc7c604c3375d3e3f572afad8d27eb06a98a 99e9af63c5ee3ed2b5d20cb910eb29596ccb8391f06c376083247fc940598bd3d888 bc579a9496eba784dfea823a0a1a28cecc20951e4bef35596147f53d4957fa965e07 1d5a5c80b982ea26b6500b9a63e2a1c1196cde4deb61a0358ef822a0e00849e807b1 10d036d268480089e21d4f07ce43d9fcf233ec0e7044460410a6cc7254becfb27d31 679a63e4285632d7477561e4af332f2cc622443f8a94e7f03a93c8ca4b10871a562a d9f6a00e9f70c273fd4bef730201d3bcd75e2bbfeac22d98c6667d7b8d0cb965b94d 1be33c5329821a6292239ac93a896d7086e435c249c484a66bf002b1b99a2f633bc8 fb9ffc8c7aeb4ed95301f4bc51441802ca28f56257e77efb556ea385e086c6e9c3fe 901b9bfa034ef9a5202d29030e5e962635baf5b878cc2b7f7414e6b68fcfd2606608 4e936eed103a4073bca9cfb6a209251aa59fc6caead3faef33ed547d47876656a55c ebb5cb6cba8b294081540f627f2235588f69eef2c4b029781a31a5884f20bd3699ff 71a0726ee9a9e41caa19bd9ad50660dd9cda8852f2c555d1f4162e13ab0dfa63d666 1f28d856838ef093286447c277aba2b4d05a6559f1f214bc73bed5f47e1e73465c58 718904618ebcb16b6a3feb719c2f3ff4e5409d1aece9aca7bba3019d5def920e2b78 512f74e44326d23a42f78b7eece89296ed9315d7bc097eace97dc3691aad6b3d7d79 885d4e1def2800e6f3e94685e66b234daeef3d2d6c638e961408892941cafed16ccd f523a91d2ae4b871250ed0b9b53e5e0e93837ca400f9d5f11030ef1536e39592a23e e10ec0f568cc373b1d65fe424cc9382fb653fb5f347acc6dab0998ee6f9f9d26bf93 c6909d88fb0b0d05298c7424a7d30dc3f54b1bbb2b9d33903b3faf133f21aa723491 7269fefe6a2691526dfa7647bce8758598524c2532b695eb174ab4c08405a7532374 167317eef95d7b2044e6409e2bdc70b9463b5a50c8647692b334904e5906405766c7 f3f6fb5e07786c6fa37c2e8d96dcf0a3b65ed4d4f031b627920586b8c23092dcc1f9 9e6267ef25b0a75a00a48070a92032e9ed0a58eb7adc4b4571e85165b50449b44414 16e550ecd612b55ef2cc6a3f8036351b00db614c248c11a1a8b7d945ed719f2fb0d0 855cea744cb637e3e709a55274d9f41b9a9026a857c1c6c30257391e5b510cf06254 fcb85c3992634a50eba9f58974de34dfa65dceca9ac467b4efa9706fb5dc196d277f 07cd46e7431b785591c39387f6671542d9ab5caf9ef97aa43f7b46bb75a4fc8661b1 4513f13609bae9722f3b4b0d03d039e5e1ae3cddd29e729e6afa7ab605be00f7dae1 3d874349affad716385139c14f90296b2bab2bef5ac8a1086660701bd4b574e9c3cc 1b5999a3f5eb67ec10c1ab025a4d621955010e44dca1a08032f12478c65755678da2 6e411890e6bcddb30c1bce42d40d036c390e6a8a63ecbd6b1b6ca3be45d4dcc82bb8 c4b3c05a66ef039c2c22f733a6531a5cf8374db73ab67b4c50d4efd000e6346f345f 1c469813505683f67ebb120c9ce5f96d1b3476ddc8e320dde1ca0e0ccfba2f292219 5b4f73defcd2dc45b7f72a5f2cba86fc0b67587d317944acc3cee8f7461571137690 bca0c62f4df293348c1cb86e41abf55d90acdb1862bac1378b1f3e04b3d0a3fa8a28 115b2ec08103eb70e0bbd8c9c6a1b6597d06e14dab1b20b6a60020f061c1ab5380e4 00b27a583c22ea32a7b481dd855c2a26096020c40e0d0f20ed75052afb7f6bebe83a 794ea0662e149306d0b03083297ef6ac920a065805d96232715ca161271c33370a71 835bbe31b722b04168a20012953f425ced6ca60fe9926a510b879376508008c8bcb1 af5a97064c2309c809ff153ff242267b44dd0075e4bd9e0ece58516cc41fda213f21 a6cb40ad2f4395ccd100f985eaaada93cf9e3964e9ca23f6f9b5ae8d72a40a4d534e a6fea998a2726615c72d86757a6c5946c45871850709855d5bd3c1683b476e21fd3a fdffe94eca83e48c283fc8462f1a618481676ff987e12fc63a19feafcf54aa9754cc 39511c2750eb0ed6ec6e059810d3f0a49f7fd47830450b1d26d4db70a18d1d76e887 7dc24e1e89bb448cdf24d5664b2228bd4c8bfb6f344ac1f1c3f535ad6dc77641bad1 88d67eedbe34b42389708d6007bdd591a5dec58663f0c75c956d7e8dd1552639d831 3a5ddbf10aefbd2f1f153971f0ad1ed0a767c9ab66d4deffb1271eb4760d280899cd afec6de7d13cd298cddf4e8450be0e7d63c9afa51139c9cc3c780557f659e37301e0 f268ff4811f8c04cd65adf3d676cee7e9c3fac68d792edb1af47edd08460fec81ae2 9dbeb3c3b4fe17f87be656102b4e654b67bff33b4e60e04171d8cd0795d58a93914d f26b6f6bcd94838e9f563e19202d2ebd12544ac97a15ba32db7d8f449f36d512622e

```
4054364c1e68df0f48688a6f9d66f3caca4651a4811d6562ebcb624e22d7bd8468e7
aa4451883b9e5421c4b849e07a52d5c4e686f19ebae0c75df57375d2f56bde33c6c9
715a4cf6f1272a0f18e88ff99eb9f79b5e48a20e1bd99a719b724c96d2c8ca07fb16
45aed79289dde7586aa9a8e219e02872c1f40c0107a2290ea950b3893273358f0230
1594d918a527555a8018786d357dfb348ce626c17cafb9b88905678b120a8d6c54a9
840e99dfb56d22e12ff7f09c2883058cfa106b54a289e02be522a0a4d19178fd7723
83ed9cfe3433bcd20c4f512d01d44ee734c71acae9785fef664de661535d1c1e4ae5
9bbb702727d07567c4266023072e3ec300322300dc601f6de9cb009bd9df68d0b54f
4adb2951af99b8d45d157d78b9bb908c079509ba3d3680e4cc4ed1ce990ba9a261cd
d1064a7755c18b67293350d2cc57a28ef1b9432326483b9c2020a1a7bf07178528b7
fcc0072eeb31019ba0cac461a55c46f7a30dd1554e2861ceb98ca52d0106c09bbfd6
e2d9c2061d797199846119294f293679740bfb06626974f93382e9e854df29bc8aef
4dc3bc1122dbabd3cc1ae646bafa5d5b5824fffc22860e737419800f8037f3d0ac05
bd041345e94e597817ff15a5f2332b697a97507954a664eed19e925daba33c575773
ebdbaf42babf42d17073e1d595fb612519ccc560451c66c82cf2f2b220d58fd6d666
a089b843f993c784e4b07027ccbb72437a69ce1b8e34050b254d3f9d195750edfaf9
2f5cff1140ea202cbc489b32ab4d78f6cbc7205cdd9ec64db4dba9477e141f1bb54a
af929e9fb766711eaa7b802818e03c2576bc5fbd54a8daf64fa8907a50b81db4b53f
87ad894342ba54304ff013586aadba30f40f8c62c127287f2ef26303fbb0be86f581
f51881686d3ee672a567c887c88122eddf9ea2e75ce4788b8b07b7bf30bd68ab766d
336d3de479d3b7b738873b53fb367af51e26e098d9f60b2ff9eb364db4847adb19f4
5a6644298dda56b1fa62857a55aeed02fd4b7ad11dabf45cc94b92e797c4d8e53646
da4847e0f84372466d6503cea856f000274afd00fd97b69d185fc14ab25a5cb9d897
b85f5e22f8f881b113b60e89b05f2717fe29a8c3eccba97755e827ef02cd95ad52a1
de25ae6deefa3f150c61f39256f3dc2959c4f0ad94ccab130c9f69642534484ba827
33bfc102aaffbd86103975cfb8146df3ffe895e462fabac9ab8e49ba900e7ddb63fd
32e2b6487fe52d1e2c42ec16da300214992fc45dbeb39474e2edce031c153c0ff88d
970c71707bde5eeb5a8cec19b79558d8cf6ba7d1fd6c6e6361fc4f4245850166cb3e
d529ba0f75c9eab38298cbbc1d365644633ad610c8bb4c82bf5d6f9eed300df18d84
f0244965c8df4df535f6e0e63dea6fe49e439f97aad7c74acc77887a35aeb46e4f7c
b648dd53b96350913d14c97106a528df78de8166c37dfc175eb8cd290b4eeafe4c03
47c2f691e41d30e0aceb7eb25d58aa81b05fe8f5300a1e38dee028cdfe196210fbe6
aa2cdcd1b426aafcf3bf086e1d34f930a1bb9ee32b575f4ae3b82ef52f44d8654353
9c358a7079b18c7178635fc1fb54f7ed6b51ad603293d21523f62b4d1978ac4014d5
07268c830c621995daf0a8937a7a820ef6ddba82a53503bd8c2e48d0863ecdfc28c4
fe15029337e1c7caf29e5dc05d447"/>
      </OpaqueObject>
    </RequestPayload>
  </BatchItem>
</RequestMessage>
```

```
<ResponseMessage>
  <ResponseHeader>
    <ProtocolVersion>
      <ProtocolVersionMajor type="Integer" value="1"/>
      <ProtocolVersionMinor type="Integer" value="0"/>
    </ProtocolVersion>
    <TimeStamp type="DateTime" value="2012-04-27T08:12:24+00:00"/>
    <BatchCount type="Integer" value="1"/>
  </ResponseHeader>
  <BatchItem>
    <Operation type="Enumeration" value="Register"/>
    <ResultStatus type="Enumeration" value="Success"/>
    <ResponsePayload>
      <UniqueIdentifier type="TextString"
value="$UNIQUE_IDENTIFIER_0"/>
    </ResponsePayload>
  </BatchItem>
</ResponseMessage>
```

```
        # TIME 1
0046    <RequestMessage>
0047      <RequestHeader>
0048        <ProtocolVersion>
0049          <ProtocolVersionMajor type="Integer" value="1"/>
0050          <ProtocolVersionMinor type="Integer" value="0"/>
0051        </ProtocolVersion>
0052        <BatchCount type="Integer" value="1"/>
0053      </RequestHeader>
0054      <BatchItem>
0055        <Operation type="Enumeration" value="Destroy"/>
0056        <RequestPayload>
0057          <UniqueIdentifier type="TextString"
        value="$UNIQUE_IDENTIFIER_0"/>
0058        </RequestPayload>
0059      </BatchItem>
0060    </RequestMessage>
```

```
0061    <ResponseMessage>
0062      <ResponseHeader>
0063        <ProtocolVersion>
0064          <ProtocolVersionMajor type="Integer" value="1"/>
0065          <ProtocolVersionMinor type="Integer" value="0"/>
0066        </ProtocolVersion>
0067        <TimeStamp type="DateTime" value="2012-04-27T08:12:24+00:00"/>
0068        <BatchCount type="Integer" value="1"/>
0069      </ResponseHeader>
0070      <BatchItem>
0071        <Operation type="Enumeration" value="Destroy"/>
0072        <ResultStatus type="Enumeration" value="Success"/>
0073        <ResponsePayload>
0074          <UniqueIdentifier type="TextString"
        value="$UNIQUE_IDENTIFIER_0"/>
0075        </ResponsePayload>
0076      </BatchItem>
0077    </ResponseMessage>
```

151

152

153 ## 3.5 Optional Test Cases KMIP 1.1

154 This section documents the test cases that a client or server conformant to the Opaque Managed Object
155 Store Profile SHALL support under KMIP Specification 1.1.

156 ### 3.5.1 OMOS-O-1-11

157 Register larger (>10k) opaque object

```
        # TIME 0
0001    <RequestMessage>
0002      <RequestHeader>
0003        <ProtocolVersion>
0004          <ProtocolVersionMajor type="Integer" value="1"/>
0005          <ProtocolVersionMinor type="Integer" value="1"/>
0006        </ProtocolVersion>
0007        <BatchCount type="Integer" value="1"/>
0008      </RequestHeader>
0009      <BatchItem>
0010        <Operation type="Enumeration" value="Register"/>
```

```
0011      <RequestPayload>
0012        <ObjectType type="Enumeration" value="OpaqueObject"/>
0013        <TemplateAttribute>
0014          <Attribute>
0015            <AttributeName type="TextString" value="Name"/>
0016            <AttributeValue>
0017              <NameValue type="TextString" value="OMOS-O-1-11"/>
0018              <NameType type="Enumeration"
      value="UninterpretedTextString"/>
0019            </AttributeValue>
0020          </Attribute>
0021        </TemplateAttribute>
0022        <OpaqueObject>
0023          <OpaqueDataType type="Enumeration" value="0x80000001"/>
0024          <OpaqueDataValue type="ByteString"
      value="168392816fd71b3d1c5d9cecfacf61f4e396374ede655d9d15305d6a0a04e
      5f0beab1de8be60fb716de00456c0b4adaadd5e1f4e72879251dbf7d25ca9f81076d
      aa0b6464ae989a76a6f6710ea9560a60b99cb4f697cd075cd799cb7dbcfffab4c2ab
      a5a19529f14307f6d217b1c84114eab50855b623d2e2a7602cdd230778939cce2a03
      550b0e0c9a4ff7e0ad2af805a92bbe4a41ba3405565ca050c38c6d5b92d902c30544
      b1460e2360459ee2ef3376b66caf91e0e0980d12ea6c19b5623cf03ad065652cf247
      ee2be155deacfda3d96b35f21d2f97fe4fd28244dec67f61c32250f5fc93dc515c1b
      5c7004f212b7c1d60972f3aa0372789364a3a762f80fda1d58389ea3cb3d204db887
      b0db62623350d4ea7d1bfd91e6d522fab6942abe5ab9f76278e4cb280fed409268c2
      731552c8292829a47355852d5780388a4e13691f8ec654226ce52e213fff30b0b3de
      7ffb7444c7748f7e90dd893276d526a657bdf42ea588721788feb605e5d3443ebe06
      91be98af902a3d6a459f1e160df7dc3a7507b05a238d49c6d5ef6803ffb964cd813d
      b90f549c2393fae94fcfc8c05ddb62a71bfc031074f4d32ada48491c970dedf57c13
      9cb04c94112fcef3eec9fdd7487eecd1470741f780e0d9e99ba68e97945b7ab7970f
      8003f80ca9622c94192281c13380894dc1f6c6d88848ffe81fd994862d2c60db1b65
      1dbf12a245d34fc0e2a1b7cf36428c1e481890607a4df45ea20619ea02946af0c7c4
      1fc16bc620159871659c8105506fb0d4017921ea79ac082afad5cb9bf703a49ac79f
      d1f428fdca5a8f693990bfbca9640a44bf43e5786111f624369d1d33a08cd7247be0
      7f9bad26e531a2f3f9aea66536682296348cf86291af9c2521bc6196747986d02b3a
      465dfbbe9468c3b364a8048441e32dc8ff190f0ad62b2c0f6a6d4aa715580b1fb2ca
      b038560259981eee6f59f850c076c29507e9efd9cca183f5bd8a0a876820ed173f88
      4f6a9773c8102859b59b286002147b86428c18537590ff9b3cbe3dd2f7607f2b7d84
      227c5d9ca6e6f272631672960f0da9c69d6b31eb499d50d724e3e3d4f0c982424389
      15de54f1a600acd13f9ce483c01dcd8c5d36a7cc57ead5f4066b849b8a5e00cc5601
      9bebb2e8129a6481e9f4b234154808236538c500ccd597a273068dc442f3c1205005
      0209538cb1ea3194a18278b55d589d34e7680f565ed411359eaddf12c0bdfe907b16
      377827d6ca46460d19f703c7b17fd3b7be9f45c162a34502f507673d304487e11c27
      c9827cbc9cc5e3992514aa6e62d3163e338af19e99fcddec6ee17f70af7c9644a297
      af3ec237619b8580bd20ae8ada4ce7dfe834711560b598f5aab65eac8fd2a8d66f57
      4c79e4f19faa08dda5823ddc4f532f3053841dc52213dc147b325599a4b5969fa264
      2fb50c01030b14253d0f23fd34f663581e95da9bfb0e3f52a010b2f5911ff063ac2a
      826c94a8789445bc229a1ef1fe74fd8b5f672e8ffa671a5c69d19a4d7611c149cd52
      68d590788aa3e44e949beb46f38a8fa51a301824e88c220ae41ba4b036c672342c04
      3ebc6db91035b70ae68d58f558a3a5d1a788a6694c4f74278a204743fae6d947b502
      e29552ba65507e91b684a3955aafb43530f02ccccaeded9a6d1f877da2470f36c260
      14653493c970613f25de9c21f0682062881de0fc6ce2712e6f924408e7d29d368c43
      ed198f14f5e947bb2f721084e6c22f750a6cd2400c49b9689e4e0f5d3d52005c5e42
      a6aa0ecdc237f7b1868b7e77bce2c0d8f160a061f0de6529967e82586fe854bc89ed
      68dc7d9ac521aa2f40e1c4c3835ad8c2881ec579975d4cf9d9beb69d0f2c4b4f1e69
      ace6da5b9f6e49801ac9e1c5922176e3c72b8f0b73629d457a10456d9c8f0ae56e40
      cb01f64ba0b49ea69f23728cffa532fd01b966ff31c30fedc9a52b7b0a5fe6a2e3bd
      53c87560cf74b12143696e52343317c408d8e13bdb1fbd2758aaa3cf8aa3fd229bb6
      5a9e41228372889147470cee02ae4acb89d52e31f8da66543fc8a6c02a6b337ce10d
```

	0f41f464a44509c245476be1e5b11df8dbb867664cec0882beee6e21a022abcb89de d8a17cf6f17da33c094e42be30a3c070950ff300993abb40fac22f0ecf7d155c9261 516796cb1de4c0d6bfc21e716b04d23dbb8b465708204a3b96af8e46042e8205e7df 92b9e01c7794c94a5036b7f85fe0967515cc05ea460c7810c0c551d8fd94f7ad7d1a 641e73a1471e9364bf4e2b4687c7700381d46d39a3159f486d0323de59c5f555d323 e5c072fbfc0758e22aa6a1df04d13bf4bfe6632853424984d56d16ed61402a640c7e 1b07d9439d1fdc2df147df02639575b50da6cd769cb367dbfe3316e03939e85880f8 b19f2689d5504f25200560dd815fd4535a1b5f70c8332c95c9d292c75e971df28ec6 ff6d70a52ad78d236a51ec4a12243a650285fbd6aa4632fab8fa56eb26f638855147 d72efcdd8a0b4367122f8e2210d39ffc87ac29ab5c5f1226bea04693e0b5c671269d 96ac0c665fbd4f5fdb04ffcd7b76f0f8c1960e7a47c88226d6bd360e4dd65d70ed68 7e2fea04822f7c8394007ee085a9362a35506696c44e531786472a3db4bddd2b63ae 1a448d0442d11f28dfed4de000c820e40174e216d274ab1321be6f21d1e4eb7e2243 0c131b2050c1fcb9ebb2823dd6fb4f4972ea4167795f911fb8a3c7f14ecf71a0e675 657263c2f4b5eb72369d9d2d457999dc15392ccc10d98308830dd9b64d95427381af a9549c5df5081de88849126197154968d96797c573901e80bc638bc4126c01dea36a 56c1afe01021a21a776f6b4e375dd42156cce98998bd3036401550b501fac4ad653b cf098db8f6d9ef76429a60137e0f507c67b57ddf829c5ed88f4369a6cc7287683fa2 25515697cc2e43ffd8108e7f1564736d043b6323ae17bda3031fab7712f886ac12af beb2686700c1133017f64363293f93ab3c4e096aff76751377e5b6e5ead512a2f3d3 6635fcac28bf7fcf5c565bcf51bf650e7e3c80194a34b7348e7517fd4301f9986cdc bbe27392291da368ca699d90bca1b49122b20649f6b95529a72ba8217546042cb975 720b3c6e08ea2ed9faf23f975524a3857f5d20f76f6df0b5cf16fe7b054f4c996042 728c8b41326f7edf94fe04b0c084762ecc7c604c3375d3e3f572afad8d27eb06a98a 99e9af63c5ee3ed2b5d20cb910eb29596ccb8391f06c376083247fc940598bd3d888 bc579a9496eba784dfea823a0a1a28cecc20951e4bef35596147f53d4957fa965e07 1d5a5c80b982ea26b6500b9a63e2a1c1196cde4deb61a0358ef822a0e00849e807b1 10d036d268480089e21d4f07ce43d9fcf233ec0e7044460410a6cc7254becfb27d31 679a63e4285632d7477561e4af332f2cc622443f8a94e7f03a93c8ca4b10871a562a d9f6a00e9f70c273fd4bef730201d3bcd75e2bbfeac22d98c6667d7b8d0cb965b94d 1be33c5329821a6292239ac93a896d7086e435c249c484a66bf002b1b99a2f633bc8 fb9ffc8c7aeb4ed95301f4bc51441802ca28f56257e77efb556ea385e086c6e9c3fe 901b9bfa034ef9a5202d29030e5e962635baf5b878cc2b7f7414e6b68fcfd2606608 4e936eed103a4073bca9cfb6a209251aa59fc6caead3faef33ed547d47876656a55c ebb5cb6cba8b294081540f627f2235588f69eef2c4b029781a31a5884f20bd3699ff 71a0726ee9a9e41caa19bd9ad50660dd9cda8852f2c555d1f4162e13ab0dfa63d666 1f28d856838ef093286447c277aba2b4d05a6559f1f214bc73bed5f47e1e73465c58 718904618ebcb16b6a3feb719c2f3ff4e5409d1aece9aca7bba3019d5def920e2b78 512f74e44326d23a42f78b7eece89296ed9315d7bc097eace97dc3691aad6b3d7d79 885d4e1def2800e6f3e94685e66b234daeef3d2d6c638e961408892941cafed16ccd f523a91d2ae4b871250ed0b9b53e5e0e93837ca400f9d5f11030ef1536e39592a23e e10ec0f568cc373b1d65fe424cc9382fb653fb5f347acc6dab0998ee6f9f9d26bf93 c6909d88fb0b0d05298c7424a7d30dc3f54b1bbb2b9d33903b3faf133f21aa723491 7269fefe6a2691526dfa7647bce8758598524c2532b695eb174ab4c08405a7532374 167317eef95d7b2044e6409e2bdc70b9463b5a50c8647692b334904e5906405766c7 f3f6fb5e07786c6fa37c2e8d96dcf0a3b65ed4d4f031b627920586b8c23092dcc1f9 9e6267ef25b0a75a00a48070a92032e9ed0a58eb7adc4b4571e85165b50449b44414 16e550ecd612b55ef2cc6a3f8036351b00db614c248c11a1a8b7d945ed719f2fb0d0 855cea744cb637e3e709a55274d9f41b9a9026a857c1c6c30257391e5b510cf06254 fcb85c3992634a50eba9f58974de34dfa65dceca9ac467b4efa9706fb5dc196d277f 07cd46e7431b785591c39387f6671542d9ab5caf9ef97aa43f7b46bb75a4fc8661b1 4513f13609bae9722f3b4b0d03d039e5e1ae3cddd29e729e6afa7ab605be00f7dae1 3d874349affad716385139c14f90296b2bab2bef5ac8a1086660701bd4b574e9c3cc 1b5999a3f5eb67ec10c1ab025a4d621955010e44dca1a08032f12478c65755678da2 6e411890e6bcddb30c1bce42d40d036c390e6a8a63ecbd6b1b6ca3be45d4dcc82bb8 c4b3c05a66ef039c2c22f733a6531a5cf8374db73ab67b4c50d4efd000e6346f345f 1c469813505683f67ebb120c9ce5f96d1b3476ddc8e320dde1ca0e0ccfba2f292219 5b4f73defcd2dc45b7f72a5f2cba86fc0b67587d317944acc3cee8f7461571137690

```
bca0c62f4df293348c1cb86e41abf55d90acdb1862bac1378b1f3e04b3d0a3fa8a28
115b2ec08103eb70e0bbd8c9c6a1b6597d06e14dab1b20b6a60020f061c1ab5380e4
00b27a583c22ea32a7b481dd855c2a26096020c40e0d0f20ed75052afb7f6bebe83a
794ea0662e149306d0b03083297ef6ac920a065805d96232715ca161271c33370a71
835bbe31b722b04168a20012953f425ced6ca60fe9926a510b879376508008c8bcb1
af5a97064c2309c809ff153ff242267b44dd0075e4bd9e0ece58516cc41fda213f21
a6cb40ad2f4395ccd100f985eaaada93cf9e3964e9ca23f6f9b5ae8d72a40a4d534e
a6fea998a2726615c72d86757a6c5946c45871850709855d5bd3c1683b476e21fd3a
fdffe94eca83e48c283fc8462f1a618481676ff987e12fc63a19feafcf54aa9754cc
39511c2750eb0ed6ec6e059810d3f0a49f7fd47830450b1d26d4db70a18d1d76e887
7dc24e1e89bb448cdf24d5664b2228bd4c8bfb6f344ac1f1c3f535ad6dc77641bad1
88d67eedbe34b42389708d6007bdd591a5dec58663f0c75c956d7e8dd1552639d831
3a5ddbf10aefbd2f1f153971f0ad1ed0a767c9ab66d4deffb1271eb4760d280899cd
afec6de7d13cd298cddf4e8450be0e7d63c9afa51139c9cc3c780557f659e37301e0
f268ff4811f8c04cd65adf3d676cee7e9c3fac68d792edb1af47edd08460fec81ae2
9dbeb3c3b4fe17f87be656102b4e654b67bff33b4e60e04171d8cd0795d58a93914d
f26b6f6bcd94838e9f563e19202d2ebd12544ac97a15ba32db7d8f449f36d512622e
4054364c1e68df0f48688a6f9d66f3caca4651a4811d6562ebcb624e22d7bd8468e7
aa4451883b9e5421c4b849e07a52d5c4e686f19ebae0c75df57375d2f56bde33c6c9
715a4cf6f1272a0f18e88ff99eb9f79b5e48a20e1bd99a719b724c96d2c8ca07fb16
45aed79289dde7586aa9a8e219e02872c1f40c0107a2290ea950b3893273358f0230
1594d918a527555a8018786d357dfb348ce626c17cafb9b88905678b120a8d6c54a9
840e99dfb56d22e12ff7f09c2883058cfa106b54a289e02be522a0a4d19178fd7723
83ed9cfe3433bcd20c4f512d01d44ee734c71acae9785fef664de661535d1c1e4ae5
9bbb702727d07567c4266023072e3ec300322300dc601f6de9cb009bd9df68d0b54f
4adb2951af99b8d45d157d78b9bb908c079509ba3d3680e4cc4ed1ce990ba9a261cd
d1064a7755c18b67293350d2cc57a28ef1b9432326483b9c2020a1a7bf07178528b7
fcc0072eeb31019ba0cac461a55c46f7a30dd1554e2861ceb98ca52d0106c09bbfd6
e2d9c2061d797199846119294f293679740bfb06626974f93382e9e854df29bc8aef
4dc3bc1122dbabd3cc1ae646bafa5d5b5824fffc22860e737419800f8037f3d0ac05
bd041345e94e597817ff15a5f2332b697a97507954a664eed19e925daba33c575773
ebdbaf42babf42d17073e1d595fb612519ccc560451c66c82cf2f2b220d58fd6d666
a089b843f993c784e4b07027ccbb72437a69ce1b8e34050b254d3f9d195750edfaf9
2f5cff1140ea202cbc489b32ab4d78f6cbc7205cdd9ec64db4dba9477e141f1bb54a
af929e9fb766711eaa7b802818e03c2576bc5fbd54a8daf64fa8907a50b81db4b53f
87ad894342ba54304ff013586aadba30f40f8c62c127287f2ef26303fbb0be86f581
f51881686d3ee672a567c887c88122eddf9ea2e75ce4788b8b07b7bf30bd68ab766d
336d3de479d3b7b738873b53fb367af51e26e098d9f60b2ff9eb364db4847adb19f4
5a6644298dda56b1fa62857a55aeed02fd4b7ad11dabf45cc94b92e797c4d8e53646
da4847e0f84372466d6503cea856f000274afd00fd97b69d185fc14ab25a5cb9d897
b85f5e22f8f881b113b60e89b05f2717fe29a8c3eccba97755e827ef02cd95ad52a1
de25ae6deefa3f150c61f39256f3dc2959c4f0ad94ccab130c9f69642534484ba827
33bfc102aaffbd86103975cfb8146df3ffe895e462fabac9ab8e49ba900e7ddb63fd
32e2b6487fe52d1e2c42ec16da300214992fc45dbeb39474e2edce031c153c0ff88d
970c71707bde5eeb5a8cec19b79558d8cf6ba7d1fd6c6e6361fc4f4245850166cb3e
d529ba0f75c9eab38298cbbc1d365644633ad610c8bb4c82bf5d6f9eed300df18d84
f0244965c8df4df535f6e0e63dea6fe49e439f97aad7c74acc77887a35aeb46e4f7c
b648dd53b96350913d14c97106a528df78de8166c37dfc175eb8cd290b4eeafe4c03
47c2f691e41d30e0aceb7eb25d58aa81b05fe8f5300a1e38dee028cdfe196210fbe6
aa2cdcd1b426aafcf3bf086e1d34f930a1bb9ee32b575f4ae3b82ef52f44d8654353
9c358a7079b18c7178635fc1fb54f7ed6b51ad603293d21523f62b4d1978ac4014d5
07268c830c621995daf0a8937a7a820ef6ddba82a53503bd8c2e48d0863ecdfc28c4
fe15029337e1c7caf29e5dc05d447"/>
0025        </OpaqueObject>
0026      </RequestPayload>
0027    </BatchItem>
0028  </RequestMessage>
0029  <ResponseMessage>
```

```
0030    <ResponseHeader>
0031      <ProtocolVersion>
0032        <ProtocolVersionMajor type="Integer" value="1"/>
0033        <ProtocolVersionMinor type="Integer" value="1"/>
0034      </ProtocolVersion>
0035      <TimeStamp type="DateTime" value="2012-04-27T08:12:24+00:00"/>
0036      <BatchCount type="Integer" value="1"/>
0037    </ResponseHeader>
0038    <BatchItem>
0039      <Operation type="Enumeration" value="Register"/>
0040      <ResultStatus type="Enumeration" value="Success"/>
0041      <ResponsePayload>
0042        <UniqueIdentifier type="TextString"
        value="$UNIQUE_IDENTIFIER_0"/>
0043      </ResponsePayload>
0044    </BatchItem>
0045  </ResponseMessage>
```

```
        # TIME 1
0046  <RequestMessage>
0047    <RequestHeader>
0048      <ProtocolVersion>
0049        <ProtocolVersionMajor type="Integer" value="1"/>
0050        <ProtocolVersionMinor type="Integer" value="1"/>
0051      </ProtocolVersion>
0052      <BatchCount type="Integer" value="1"/>
0053    </RequestHeader>
0054    <BatchItem>
0055      <Operation type="Enumeration" value="Destroy"/>
0056      <RequestPayload>
0057        <UniqueIdentifier type="TextString"
        value="$UNIQUE_IDENTIFIER_0"/>
0058      </RequestPayload>
0059    </BatchItem>
0060  </RequestMessage>
```

```
0061  <ResponseMessage>
0062    <ResponseHeader>
0063      <ProtocolVersion>
0064        <ProtocolVersionMajor type="Integer" value="1"/>
0065        <ProtocolVersionMinor type="Integer" value="1"/>
0066      </ProtocolVersion>
0067      <TimeStamp type="DateTime" value="2012-04-27T08:12:24+00:00"/>
0068      <BatchCount type="Integer" value="1"/>
0069    </ResponseHeader>
0070    <BatchItem>
0071      <Operation type="Enumeration" value="Destroy"/>
0072      <ResultStatus type="Enumeration" value="Success"/>
0073      <ResponsePayload>
0074        <UniqueIdentifier type="TextString"
        value="$UNIQUE_IDENTIFIER_0"/>
0075      </ResponsePayload>
0076    </BatchItem>
0077  </ResponseMessage>
```

158

159

3.6 Optional Test Cases KMIP 1.2

~~This section documents the test cases that a client or server conformant to the Opaque Managed Object Store Profile MAY support under KMIP Specification 1.2.~~

3.6.1 OMOS-O-1-12

Register larger (>10k) opaque object

```
# TIME 0
<RequestMessage>
  <RequestHeader>
    <ProtocolVersion>
      <ProtocolVersionMajor type="Integer" value="1"/>
      <ProtocolVersionMinor type="Integer" value="2"/>
    </ProtocolVersion>
    <BatchCount type="Integer" value="1"/>
  </RequestHeader>
  <BatchItem>
    <Operation type="Enumeration" value="Register"/>
    <RequestPayload>
      <ObjectType type="Enumeration" value="OpaqueObject"/>
      <TemplateAttribute>
        <Attribute>
          <AttributeName type="TextString" value="Name"/>
          <AttributeValue>
            <NameValue type="TextString" value="OMOS-O-1-12"/>
            <NameType type="Enumeration"
value="UninterpretedTextString"/>
          </AttributeValue>
        </Attribute>
      </TemplateAttribute>
      <OpaqueObject>
        <OpaqueDataType type="Enumeration" value="0x80000001"/>
        <OpaqueDataValue type="ByteString"
value="168392816fd71b3d1c5d9cecfacf61f4e396374ede655d9d15305d6a0a04e
5f0beab1de8be60fb716de00456c0b4adaadd5e1f4e72879251dbf7d25ca9f81076d
aa0b6464ae989a76a6f6710ea9560a60b99cb4f697cd075cd799cb7dbcfffab4c2ab
a5a19529f14307f6d217b1c84114eab50855b623d2e2a7602cdd230778939cce2a03
550b0e0c9a4ff7e0ad2af805a92bbe4a41ba3405565ca050c38c6d5b92d902c30544
b1460e2360459ee2ef3376b66caf91e0e0980d12ea6c19b5623cf03ad065652cf247
ee2be155deacfda3d96b35f21d2f97fe4fd28244dec67f61c32250f5fc93dc515c1b
5c7004f212b7c1d60972f3aa0372789364a3a762f80fda1d58389ea3cb3d204db887
b0db62623350d4ea7d1bfd91e6d522fab6942abe5ab9f76278e4cb280fed409268c2
731552c8292829a47355852d5780388a4e13691f8ec654226ce52e213fff30b0b3de
7ffb7444c7748f7e90dd893276d526a657bdf42ea588721788feb605e5d3443ebe06
91be98af902a3d6a459f1e160df7dc3a7507b05a238d49c6d5ef6803ffb964cd813d
b90f549c2393fae94fcfc8c05ddb62a71bfc031074f4d32ada48491c970dedf57c13
9cb04c94112fcef3eec9fdd7487eecd1470741f780e0d9e99ba68e97945b7ab7970f
8003f80ca9622c94192281c13380894dc1f6c6d88848ffe81fd994862d2c60db1b65
1dbf12a245d34fc0e2a1b7cf36428c1e481890607a4df45ea20619ea02946af0c7c4
1fc16bc620159871659c8105506fb0d4017921ea79ac082afad5cb9bf703a49ac79f
d1f428fdca5a8f693990bfbca9640a44bf43e5786111f624369d1d33a08cd7247be0
7f9bad26e531a2f3f9aea66536682296348cf86291af9c2521bc6196747986d02b3a
465dfbbe9468c3b364a8048441e32dc8ff190f0ad62b2c0f6a6d4aa715580b1fb2ca
b038560259981eee6f59f850c076c29507e9efd9cca183f5bd8a0a876820ed173f88
4f6a9773c8102859b59b286002147b86428c18537590ff9b3cbe3dd2f7607f2b7d84
227c5d9ca6e6f272631672960f0da9c69d6b31eb499d50d724e3e3d4f0c982424389
15de54f1a600acd13f9ce483c01dcd8c5d36a7cc57ead5f4066b849b8a5e00cc5601
```

	9bebb2e8129a6481e9f4b234154808236538c500ccd597a273068dc442f3c1205005 0209538cb1ea3194a18278b55d589d34e7680f565ed411359eaddf12c0bdfe907b16 377827d6ca46460d19f703c7b17fd3b7be9f45c162a34502f507673d304487e11c27 c9827cbc9cc5e3992514aa6e62d3163e338af19e99fcddec6ee17f70af7c9644a297 af3ec237619b8580bd20ae8ada4ce7dfe834711560b598f5aab65eac8fd2a8d66f57 4c79e4f19faa08dda5823ddc4f532f3053841dc52213dc147b325599a4b5969fa264 2fb50c01030b14253d0f23fd34f663581e95da9bfb0e3f52a010b2f5911ff063ac2a 826c94a8789445bc229a1ef1fe74fd8b5f672e8ffa671a5c69d19a4d7611c149cd52 68d590788aa3e44e949beb46f38a8fa51a301824e88c220ae41ba4b036c672342c04 3ebc6db91035b70ae68d58f558a3a5d1a788a6694c4f74278a204743fae6d947b502 e29552ba65507e91b684a3955aafb43530f02ccccaeded9a6d1f877da2470f36c260 14653493c970613f25de9c21f0682062881de0fc6ce2712e6f924408e7d29d368c43 ed198f14f5e947bb2f721084e6c22f750a6cd2400c49b9689e4e0f5d3d52005c5e42 a6aa0ecdc237f7b1868b7e77bce2c0d8f160a061f0de6529967e82586fe854bc89ed 68dc7d9ac521aa2f40e1c4c3835ad8c2881ec579975d4cf9d9beb69d0f2c4b4f1e69 ace6da5b9f6e49801ac9e1c5922176e3c72b8f0b73629d457a10456d9c8f0ae56e40 cb01f64ba0b49ea69f23728cffa532fd01b966ff31c30fedc9a52b7b0a5fe6a2e3bd 53c87560cf74b12143696e52343317c408d8e13bdb1fbd2758aaa3cf8aa3fd229bb6 5a9e41228372889147470cee02ae4acb89d52e31f8da66543fc8a6c02a6b337ce10d 0f41f464a44509c245476be1e5b11df8dbb867664cec0882beee6e21a022abcb89de d8a17cf6f17da33c094e42be30a3c070950ff300993abb40fac22f0ecf7d155c9261 516796cb1de4c0d6bfc21e716b04d23dbb8b465708204a3b96af8e46042e8205e7df 92b9e01c7794c94a5036b7f85fe0967515cc05ea460c7810c0c551d8fd94f7ad7d1a 641e73a1471e9364bf4e2b4687c7700381d46d39a3159f486d0323de59c5f555d323 e5c072fbfc0758e22aa6a1df04d13bf4bfe6632853424984d56d16ed61402a640c7e 1b07d9439d1fdc2df147df02639575b50da6cd769cb367dbfe3316e03939e85880f8 b19f2689d5504f25200560dd815fd4535a1b5f70c8332c95c9d292c75e971df28ec6 ff6d70a52ad78d236a51ec4a12243a650285fbd6aa4632fab8fa56eb26f638855147 d72efcdd8a0b4367122f8e2210d39ffc87ac29ab5c5f1226bea04693e0b5c671269d 96ac0c665fbd4f5fdb04ffcd7b76f0f8c1960e7a47c88226d6bd360e4dd65d70ed68 7e2fea04822f7c8394007ee085a9362a35506696c44e531786472a3db4bddd2b63ae 1a448d0442d11f28dfed4de000c820e40174e216d274ab1321be6f21d1e4eb7e2243 0c131b2050c1fcb9ebb2823dd6fb4f4972ea4167795f911fb8a3c7f14ecf71a0e675 657263c2f4b5eb72369d9d2d457999dc15392ccc10d98308830dd9b64d95427381af a9549c5df5081de88849126197154968d96797c573901e80bc638bc4126c01dea36a 56c1afe01021a21a776f6b4e375dd42156cce98998bd3036401550b501fac4ad653b cf098db8f6d9ef76429a60137e0f507c67b57ddf829c5ed88f4369a6cc7287683fa2 25515697cc2e43ffd8108e7f1564736d043b6323ae17bda3031fab7712f886ac12af beb2686700c1133017f64363293f93ab3c4e096aff76751377e5b6e5ead512a2f3d3 6635fcac28bf7fcf5c565bcf51bf650e7e3c80194a34b7348e7517fd4301f9986cdc bbe27392291da368ca699d90bca1b49122b20649f6b95529a72ba8217546042cb975 720b3c6e08ea2ed9faf23f975524a3857f5d20f76f6df0b5cf16fe7b054f4c996042 728c8b41326f7edf94fe04b0c084762ecc7c604c3375d3e3f572afad8d27eb06a98a 99e9af63c5ee3ed2b5d20cb910eb29596ccb8391f06c376083247fc940598bd3d888 bc579a9496eba784dfea823a0a1a28cecc20951e4bef35596147f53d4957fa965e07 1d5a5c80b982ea26b6500b9a63e2a1c1196cde4deb61a0358ef822a0e00849e807b1 10d036d268480089e21d4f07ce43d9fcf233ec0e7044460410a6cc7254becfb27d31 679a63e4285632d7477561e4af332f2cc622443f8a94e7f03a93c8ca4b10871a562a d9f6a00e9f70c273fd4bef730201d3bcd75e2bbfeac22d98c6667d7b8d0cb965b94d 1be33c5329821a6292239ac93a896d7086e435c249c484a66bf002b1b99a2f633bc8 fb9ffc8c7aeb4ed95301f4bc51441802ca28f56257e77efb556ea385e086c6e9c3fe 901b9bfa034ef9a5202d29030e5e962635baf5b878cc2b7f7414e6b68fcfd2606608 4e936eed103a4073bca9cfb6a209251aa59fc6caead3faef33ed547d47876656a55c ebb5cb6cba8b294081540f627f2235588f69eef2c4b029781a31a5884f20bd3699ff 71a0726ee9a9e41caa19bd9ad50660dd9cda8852f2c555d1f4162e13ab0dfa63d666 1f28d856838ef093286447c277aba2b4d05a6559f1f214bc73bed5f47e1e73465c58 718904618ebcb16b6a3feb719c2f3ff4e5409d1aece9aca7bba3019d5def920e2b78 512f74e44326d23a42f78b7eece89296ed9315d7bc097eace97dc3691aad6b3d7d79

	885d4e1def2800e6f3e94685e66b234daeef3d2d6c638e961408892941cafed16ccd f523a91d2ae4b871250ed0b9b53e5e0e93837ca400f9d5f11030ef1536e39592a23e e10ec0f568cc373b1d65fe424cc9382fb653fb5f347acc6dab0998ee6f9f9d26bf93 c6909d88fb0b0d05298c7424a7d30dc3f54b1bbb2b9d33903b3faf133f21aa723491 7269fefe6a2691526dfa7647bce8758598524c2532b695eb174ab4c08405a7532374 167317eef95d7b2044e6409e2bdc70b9463b5a50c8647692b334904e5906405766c7 f3f6fb5e07786c6fa37c2e8d96dcf0a3b65ed4d4f031b627920586b8c23092dcc1f9 9e6267ef25b0a75a00a48070a92032e9ed0a58eb7adc4b4571e85165b50449b44414 16e550ecd612b55ef2cc6a3f8036351b00db614c248c11a1a8b7d945ed719f2fb0d0 855cea744cb637e3e709a55274d9f41b9a9026a857c1c6c30257391e5b510cf06254 fcb85c3992634a50eba9f58974de34dfa65dceca9ac467b4efa9706fb5dc196d277f 07cd46e7431b785591c39387f6671542d9ab5caf9ef97aa43f7b46bb75a4fc8661b1 4513f13609bae9722f3b4b0d03d039e5e1ae3cddd29e729e6afa7ab605be00f7dae1 3d874349affad716385139c14f90296b2bab2bef5ac8a1086660701bd4b574e9c3cc 1b5999a3f5eb67ec10c1ab025a4d621955010e44dca1a08032f12478c65755678da2 6e411890e6bcddb30c1bce42d40d036c390e6a8a63ecbd6b1b6ca3be45d4dcc82bb8 c4b3c05a66ef039c2c22f733a6531a5cf8374db73ab67b4c50d4efd000e6346f345f 1c469813505683f67ebb120c9ce5f96d1b3476ddc8e320dde1ca0e0ccfba2f292219 5b4f73defcd2dc45b7f72a5f2cba86fc0b67587d317944acc3cee8f7461571137690 bca0c62f4df293348c1cb86e41abf55d90acdb1862bac1378b1f3e04b3d0a3fa8a28 115b2ec08103eb70e0bbd8c9c6a1b6597d06e14dab1b20b6a60020f061c1ab5380e4 00b27a583c22ea32a7b481dd855c2a26096020c40e0d0f20ed75052afb7f6bebe83a 794ea0662e149306d0b03083297ef6ac920a065805d96232715ca161271c33370a71 835bbe31b722b04168a20012953f425ced6ca60fe9926a510b879376508008c8bcb1 af5a97064c2309c809ff153ff242267b44dd0075e4bd9e0ece58516cc41fda213f21 a6cb40ad2f4395ccd100f985eaaada93cf9e3964e9ca23f6f9b5ae8d72a40a4d534e a6fea998a2726615c72d86757a6c5946c45871850709855d5bd3c1683b476e21fd3a fdffe94eca83e48c283fc8462f1a618481676ff987e12fc63a19feafcf54aa9754cc 39511c2750eb0ed6ec6e059810d3f0a49f7fd47830450b1d26d4db70a18d1d76e887 7dc24e1e89bb448cdf24d5664b2228bd4c8bfb6f344ac1f1c3f535ad6dc77641bad1 88d67eedbe34b42389708d6007bdd591a5dec58663f0c75c956d7e8dd1552639d831 3a5ddbf10aefbd2f1f153971f0ad1ed0a767c9ab66d4deffb1271eb4760d280899cd afec6de7d13cd298cddf4e8450be0e7d63c9afa51139c9cc3c780557f659e37301e0 f268ff4811f8c04cd65adf3d676cee7e9c3fac68d792edb1af47edd08460fec81ae2 9dbeb3c3b4fe17f87be656102b4e654b67bff33b4e60e04171d8cd0795d58a93914d f26b6f6bcd94838e9f563e19202d2ebd12544ac97a15ba32db7d8f449f36d512622e 4054364c1e68df0f48688a6f9d66f3caca4651a4811d6562ebcb624e22d7bd8468e7 aa4451883b9e5421c4b849e07a52d5c4e686f19ebae0c75df57375d2f56bde33c6c9 715a4cf6f1272a0f18e88ff99eb9f79b5e48a20e1bd99a719b724c96d2c8ca07fb16 45aed79289dde7586aa9a8e219e02872c1f40c0107a2290ea950b3893273358f0230 1594d918a527555a8018786d357dfb348ce626c17cafb9b88905678b120a8d6c54a9 840e99dfb56d22e12ff7f09c2883058cfa106b54a289e02be522a0a4d19178fd7723 83ed9cfe3433bcd20c4f512d01d44ee734c71acae9785fef664de661535d1c1e4ae5 9bbb702727d07567c4266023072e3ec300322300dc601f6de9cb009bd9df68d0b54f 4adb2951af99b8d45d157d78b9bb908c079509ba3d3680e4cc4ed1ce990ba9a261cd d1064a7755c18b67293350d2cc57a28ef1b9432326483b9c2020a1a7bf07178528b7 fcc0072eeb31019ba0cac461a55c46f7a30dd1554e2861ceb98ca52d0106c09bbfd6 e2d9c2061d7971998461192 94f293679740bfb06626974f93382e9e854df29bc8aef 4dc3bc1122dbabd3cc1ae646bafa5d5b5824fffc22860e737419800f8037f3d0ac05 bd041345e94e597817ff15a5f2332b697a97507954a664eed19e925daba33c575773 ebdbaf42babf42d17073e1d595fb612519ccc560451c66c82cf2f2b220d58fd6d666 a089b843f993c784e4b07027ccbb72437a69ce1b8e34050b254d3f9d195750edfaf9 2f5cff1140ea202cbc489b32ab4d78f6cbc7205cdd9ec64db4dba9477e141f1bb54a af929e9fb766711eaa7b802818e03c2576bc5fbd54a8daf64fa8907a50b81db4b53f 87ad894342ba54304ff013586aadba30f40f8c62c127287f2ef26303fbb0be86f581 f51881686d3ee672a567c887c88122eddf9ea2e75ce4788b8b07b7bf30bd68ab766d 336d3de479d3b7b738873b53fb367af51e26e098d9f60b2ff9eb364db4847adb19f4 5a6644298dda56b1fa62857a55aeed02fd4b7ad11dabf45cc94b92e797c4d8e53646

```
da4847e0f84372466d6503cea856f000274afd00fd97b69d185fc14ab25a5cb9d897
b85f5e22f8f881b113b60e89b05f2717fe29a8c3eccba97755e827ef02cd95ad52a1
de25ae6deefa3f150c61f39256f3dc2959c4f0ad94ccab130c9f69642534484ba827
33bfc102aaffbd86103975cfb8146df3ffe895e462fabac9ab8e49ba900e7ddb63fd
32e2b6487fe52d1e2c42ec16da300214992fc45dbeb39474e2edce031c153c0ff88d
970c71707bde5eeb5a8cec19b79558d8cf6ba7d1fd6c6e6361fc4f4245850166cb3e
d529ba0f75c9eab38298cbbc1d365644633ad610c8bb4c82bf5d6f9eed300df18d84
f0244965c8df4df535f6e0e63dea6fe49e439f97aad7c74acc77887a35aeb46e4f7c
b648dd53b96350913d14c97106a528df78de8166c37dfc175eb8cd290b4eeafe4c03
47c2f691e41d30e0aceb7eb25d58aa81b05fe8f5300a1e38dee028cdfe196210fbe6
aa2cdcd1b426aafcf3bf086e1d34f930a1bb9ee32b575f4ae3b82ef52f44d8654353
9c358a7079b18c7178635fc1fb54f7ed6b51ad603293d21523f62b4d1978ac4014d5
07268c830c621995daf0a8937a7a820ef6ddba82a53503bd8c2e48d0863ecdfc28c4
fe15029337e1c7caf29e5dc05d447"/>
0025        </OpaqueObject>
0026      </RequestPayload>
0027    </BatchItem>
0028  </RequestMessage>
```

```
0029  <ResponseMessage>
0030    <ResponseHeader>
0031      <ProtocolVersion>
0032        <ProtocolVersionMajor type="Integer" value="1"/>
0033        <ProtocolVersionMinor type="Integer" value="2"/>
0034      </ProtocolVersion>
0035      <TimeStamp type="DateTime" value="2012-04-27T08:12:24+00:00"/>
0036      <BatchCount type="Integer" value="1"/>
0037    </ResponseHeader>
0038    <BatchItem>
0039      <Operation type="Enumeration" value="Register"/>
0040      <ResultStatus type="Enumeration" value="Success"/>
0041      <ResponsePayload>
0042        <UniqueIdentifier type="TextString"
      value="$UNIQUE_IDENTIFIER_0"/>
0043      </ResponsePayload>
0044    </BatchItem>
0045  </ResponseMessage>
```

```
      # TIME 1
0046  <RequestMessage>
0047    <RequestHeader>
0048      <ProtocolVersion>
0049        <ProtocolVersionMajor type="Integer" value="1"/>
0050        <ProtocolVersionMinor type="Integer" value="2"/>
0051      </ProtocolVersion>
0052      <BatchCount type="Integer" value="1"/>
0053    </RequestHeader>
0054    <BatchItem>
0055      <Operation type="Enumeration" value="Destroy"/>
0056      <RequestPayload>
0057        <UniqueIdentifier type="TextString"
      value="$UNIQUE_IDENTIFIER_0"/>
0058      </RequestPayload>
0059    </BatchItem>
0060  </RequestMessage>
```

```
0061  <ResponseMessage>
0062    <ResponseHeader>
0063      <ProtocolVersion>
0064        <ProtocolVersionMajor type="Integer" value="1"/>
0065        <ProtocolVersionMinor type="Integer" value="2"/>
```

```
0066      </ProtocolVersion>
0067      <TimeStamp type="DateTime" value="2012-04-27T08:12:24+00:00"/>
0068      <BatchCount type="Integer" value="1"/>
0069    </ResponseHeader>
0070    <BatchItem>
0071      <Operation type="Enumeration" value="Destroy"/>
0072      <ResultStatus type="Enumeration" value="Success"/>
0073      <ResponsePayload>
0074        <UniqueIdentifier type="TextString"
          value="$UNIQUE_IDENTIFIER_0"/>
0075      </ResponsePayload>
0076    </BatchItem>
0077  </ResponseMessage>
```

165

4 Conformance

4.1 Opaque Managed Object Store Client KMIP v1.0 Profile

KMIP client ~~and server~~ implementations conformant to this profile:

1. SHALL support the Authentication Suite conditions (2.1) and;
2. SHALL support the Opaque Managed Object Store – Client ~~Baseline~~ conditions (1.1~~2.2~~) and;
3. SHALL support all Mandatory Test Cases (3.1).

4.2 Opaque Managed Object Store Client KMIP v1.1 Profile

KMIP client implementations conformant to this profile:

1. SHALL support the Authentication Suite conditions (2.1) and ~~3.2~~;
2. SHALL support the Opaque Managed Object Store – Client conditions (1.1) and ~~3.3), for each supported protocol version (major~~;
3. SHALL support all Mandatory Test Cases (3.2).

4.3 Opaque Managed Object Store Client KMIP v1.2 Profile

KMIP client implementations conformant to this profile:

~~3.~~1. SHALL support the Authentication Suite conditions (2.1) and ~~minor), returning results in accordance with the test cases.~~;
2. SHALL support the Opaque Managed Object Store – Client conditions (1.1) and;
3. SHALL support all Mandatory Test Cases (3.3).

4.4 Opaque Managed Object Store Server KMIP v1.0 Profile

KMIP server implementations conformant to this profile:

1. SHALL support the Authentication Suite conditions (2.1) and;
2. SHALL support the Opaque Managed Object Store – Server conditions (2.3) and;
3. SHALL support all Mandatory Test Cases (3.1).

4.5 Opaque Managed Object Store Server KMIP v1.1 Profile

KMIP server implementations conformant to this profile:

1. SHALL support the Authentication Suite conditions (2.1) and;
2. SHALL support the Opaque Managed Object Store – Server conditions (2.3) and;
3. SHALL support all Mandatory Test Cases (3.2).

4.6 Opaque Managed Object Store Server KMIP v1.2 Profile

KMIP server implementations conformant to this profile:

1. SHALL support the Authentication Suite conditions (2.1) and;
2. SHALL support the Opaque Managed Object Store – Server conditions (2.3) and;
3. SHALL support all Mandatory Test Cases (3.3).

4.14.7 Permitted Test Case Variations

Whilst the test cases provided in this Profile define the allowed request and response content, some inherent variations MAY occur and are permitted within a successfully completed test case.

Each test case MAY include allowed variations in the description of the test case in addition to the variations noted in this section.

Other variations not explicitly noted in this Profile SHALL be deemed non-conformant.

4.1.14.7.1 Variable Items

An implementation conformant to this Profile MAY vary the following values:

1. UniqueIdentifier
2. PrivateKeyUniqueIdentifier
3. PublicKeyUniqueIdentifier
4. UniqueBatchItemIdentifier
5. AsynchronousCorrelationValue
6. TimeStamp
7. KeyValue / KeyMaterial including:
    a. key material content returned for managed cryptographic objects which are generated by the server
    b. wrapped versions of keys where the wrapping key is dynamic or the wrapping contains variable output for each wrap operation
8. For response containing the output of cryptographic operation in Data / SignatureData/ MACData / IVCounterNonce where:
    a. the managed object is generated by the server; or
    b. the operation inherently contains variable output
9. For the following DateTime attributes where the value is not specified in the request as a fixed DateTime value:
    a. ActivationDate
    b. ArchiveDate
    c. CompromiseDate
    d. CompromiseOccurrenceDate
    e. DeactivationDate
    f. DestroyDate
    g. InitialDate
    h. LastChangeDate
    i. ProtectStartDate
    j. ProcessStopDate
    k. ValidityDate
    l. OriginalCreationDate
10. LinkedObjectIdentifier
11. DigestValue
    a. For those managed cryptographic objects which are dynamically generated
12. KeyFormatType
    a. The key format type selected by the server when it creates managed objects
13. Digest

a. The HashingAlgorithm selected by the server when it calculates the digest for a managed object for which it has access to the key material

b. The Digest Value

14. Extensions reported in Query for ExtensionList and ExtensionMap
15. Application Namespaces reported in Query
16. Object Types reported in Query other than those noted as required in this profile
17. Operation Types reported in Query other than those noted as required in this profile (or any referenced profile documents)
18. For TextString attribute values containing test identifiers:

    a. Additional vendor or application prefixes

19. Additional attributes beyond those noted in the response

An implementation conformant to this Profile MAY allow the following response variations:

~~1.~~20. Object Group values – May or may not return one or more Object Group values not included in the requests

~~2.~~21. y-CustomAttributes – May or may not include additional server-specific associated attributes not included in requests

~~3.~~22. Message Extensions – May or may not include additional (non-critical) vendor extensions

~~4.~~23. TemplateAttribute – May or may not be included in responses where the Template Attribute response is noted as optional in [KMIP-SPEC]

~~5.~~24. AttributeIndex – May or may not include Attribute Index value where the Attribute Index value is 0 for Protocol Versions 1.1 and above.

~~6.~~25. ResultMessage – May or may not be included in responses and the value (if included) may vary from the text contained within the test case.

~~7.~~26. The list of Protocol Versions returned in a DiscoverVersion response may include additional protocol versions if the request has not specified a list of client supported Protocol Versions.

~~8.~~27. VendorIdentification - The value (if included) may vary from the text contained within the test case.

~~4.1.2~~4.7.2 Variable behavior

An implementation conformant to this Profile SHALL allow variation of the following behavior:

1. A test may omit the clean-up requests and responses (containing Revoke and/or Destroy) at the end of the test provided there is a separate mechanism to remove the created objects during testing.
2. A test may omit the test identifiers if the client is unable to include them in requests. This includes the following attributes:

   a. Name; and

   b. x-ID

3. A test MAY perform requests with multiple batch items or as multiple requests with a single batch item provided the sequence of operations are equivalent
4. A request MAY contain an optional *Authentication* [KMIP_SPEC] structure within each request

Appendix A. Acknowledgments

The following individuals have participated in the creation of this specification and are gratefully acknowledged:

Participants:

Hal Aldridge, Sypris Electronics
Mike Allen, Symantec
Gordon Arnold, IBM
Todd Arnold, IBM
Richard Austin, Hewlett-Packard
Lars Bagnert, PrimeKey
Elaine Barker, NIST
Peter Bartok, Venafi, Inc.
Tom Benjamin, IBM
Anthony Berglas, Cryptsoft
Mathias Björkqvist, IBM
Kevin Bocket, Venafi
Anne Bolgert, IBM
Alan Brown, Thales e-Security
Tim Bruce, CA Technologies
Chris Burchett, Credant Technologies, Inc.
Kelley Burgin, National Security Agency
Robert Burns, Thales e-Security
Chuck Castleton, Venafi
Kenli Chong, QuintessenceLabs
John Clark, Hewlett-Packard
Tom Clifford, Symantec Corp.
Doron Cohen, SafeNet, Inc
Tony Cox, Cryptsoft
Russell Dietz, SafeNet, Inc
Graydon Dodson, Lexmark International Inc.
Vinod Duggirala, EMC Corporation
Chris Dunn, SafeNet, Inc.
Michael Duren, Sypris Electronics
James Dzierzanowski, American Express CCoE
Faisal Faruqui, Thales e-Security
Stan Feather, Hewlett-Packard
David Finkelstein, Symantec Corp.
James Fitzgerald, SafeNet, Inc.
Indra Fitzgerald, Hewlett-Packard
Judith Furlong, EMC Corporation
Susan Gleeson, Oracle
Robert Griffin, EMC Corporation
Paul Grojean, Individual
Robert Haas, IBM
Thomas Hardjono, M.I.T.
ChengDong He, Huawei Technologies Co., Ltd.
Steve He, Vormetric
Kurt Heberlein, Hewlett-Packard
Larry Hofer, Emulex Corporation
Maryann Hondo, IBM
Walt Hubis, NetApp
Tim Hudson, Cryptsoft
Jonas Iggbom, Venafi, Inc.

Sitaram Inguva, American Express CCoE
Jay Jacobs, Target Corporation
Glen Jaquette, IBM
Mahadev Karadiguddi, NetApp
Greg Kazmierczak, Wave Systems Corp.
Marc Kenig, SafeNet, Inc.
Mark Knight, Thales e-Security
Kathy Kriese, Symantec Corporation
Mark Lambiase, SecureAuth
John Leiseboer, Quintenssence Labs
Hal Lockhart, Oracle Corporation
Robert Lockhart, Thales e-Security
Anne Luk, Cryptsoft
Sairam Manidi, Freescale
Luther Martin, Voltage Security
Neil McEvoy, iFOSSF
Marina Milshtein, Individual
Dale Moberg, Axway Software
Jishnu Mukeri, Hewlett-Packard
Bryan Olson, Hewlett-Packard
John Peck, IBM
Rob Philpott, EMC Corporation
Denis Pochuev, SafeNet, Inc.
Reid Poole, Venafi, Inc.
Ajai Puri, SafeNet, Inc.
Saravanan Ramalingam, Thales e-Security
Peter Reed, SafeNet, Inc.
Bruce Rich, IBM
Christina Richards, American Express CCoE
Warren Robbins, Dell
Peter Robinson, EMC Corporation
Scott Rotondo, Oracle
Saikat Saha, SafeNet, Inc.
Anil Saldhana, Red Hat
Subhash Sankuratripati, NetApp
Boris Schumperli, Cryptomathic
Greg Singh, QuintessenceLabs
David Smith, Venafi, Inc
Brian Spector, Certivox
Terence Spies, Voltage Security
Deborah Steckroth, RouteOne LLC
Michael Stevens, QuintessenceLabs
Marcus Streets, Thales e-Security
Satish Sundar, IBM
Kiran Thota, VMware
Somanchi Trinath, Freescale Semiconductor, Inc.
Nathan Turajski, Thales e-Security
Sean Turner, IECA, Inc.
Paul Turner, Venafi, Inc.
Rod Wideman, Quantum Corporation
Steven Wierenga, Hewlett-Packard
Jin Wong, QuintessenceLabs
Sameer Yami, Thales e-Security
Peter Yee, EMC Corporation
Krishna Yellepeddy, IBM
Catherine Ying, SafeNet, Inc.
Tatu Ylonen, SSH Communications Security (Tectia Corp)

392 Michael Yoder, Vormetric. Inc.
393 Magda Zdunkiewicz, Cryptsoft
394 Peter Zelechoski, Election Systems & Software

Appendix B. KMIP Specification Cross Reference

Reference Term	KMIP 1.0	KMIP 1.1	KMIP 1.2
1 Introduction			
Non-Normative References	1.3.	1.3.	1.3.
Normative References	1.2.	1.2.	1.2.
Terminology	1.1.	1.1.	1.1.
2 Objects			
Attribute	2.1.1.	2.1.1.	2.1.1.
Base Objects	2.1.	2.1.	2.1.
Certificate	2.2.1.	2.2.1.	2.2.1.
Credential	2.1.2.	2.1.2.	2.1.2.
Data	-	-	2.1.10.
Data Length	-	-	2.1.11.
Extension Information	-	2.1.9.	2.1.9.
Key Block	2.1.3.	2.1.3.	2.1.3.
Key Value	2.1.4.	2.1.4.	2.1.4.
Key Wrapping Data	2.1.5.	2.1.5.	2.1.5.
Key Wrapping Specification	2.1.6.	2.1.6.	2.1.6.
MAC Data	-	-	2.1.13.
Managed Objects	2.2.	2.2.	2.2.
Nonce	-	-	2.1.14.
Opaque Object	2.2.8.	2.2.8.	2.2.8.
PGP Key	-	-	2.2.9.
Private Key	2.2.4.	2.2.4.	2.2.4.
Public Key	2.2.3.	2.2.3.	2.2.3.
Secret Data	2.2.7.	2.2.7.	2.2.7.
Signature Data	-	-	2.1.12.
Split Key	2.2.5.	2.2.5.	2.2.5.
Symmetric Key	2.2.2.	2.2.2.	2.2.2.
Template	2.2.6.	2.2.6.	2.2.6.
Template-Attribute Structures	2.1.8.	2.1.8.	2.1.8.
Transparent DH Private Key	2.1.7.6.	2.1.7.6.	2.1.7.6.
Transparent DH Public Key	2.1.7.7.	2.1.7.7.	2.1.7.7.
Transparent DSA Private Key	2.1.7.2.	2.1.7.2.	2.1.7.2.
Transparent DSA Public Key	2.1.7.3.	2.1.7.3.	2.1.7.3.
Transparent ECDH Private Key	2.1.7.10.	2.1.7.10.	2.1.7.10.
Transparent ECDH Public Key	2.1.7.11.	2.1.7.11.	2.1.7.11.
Transparent ECDSA Private Key	2.1.7.8.	2.1.7.8.	2.1.7.8.
Transparent ECDSA Public Key	2.1.7.9.	2.1.7.9.	2.1.7.9.
Transparent ECMQV Private Key	2.1.7.12.	2.1.7.12.	2.1.7.12.
Transparent ECMQV Public Key	2.1.7.13.	2.1.7.13.	2.1.7.13.
Transparent Key Structures	2.1.7.	2.1.7.	2.1.7.
Transparent RSA Private Key	2.1.7.4.	2.1.7.4.	2.1.7.4.
Transparent RSA Public Key	2.1.7.5.	2.1.7.5.	2.1.7.5.
Transparent Symmetric Key	2.1.7.1.	2.1.7.1.	2.1.7.1.
3 Attributes			
Activation Date	3.19.	3.24.	3.24.
Alternative Name	-	-	3.40.
Application Specific Information	3.30.	3.36.	3.36.
Archive Date	3.27.	3.32.	3.32.

Reference Term	KMIP 1.0	KMIP 1.1	KMIP 1.2
Attributes	3	3	3
Certificate Identifier	3.9.	3.13.	3.13.
Certificate Issuer	3.11.	3.15.	3.15.
Certificate Length	-	3.9.	3.9.
Certificate Subject	3.10.	3.14.	3.14.
Certificate Type	3.8.	3.8.	3.8.
Compromise Date	3.25.	3.30.	3.30.
Compromise Occurrence Date	3.24.	3.29.	3.29.
Contact Information	3.31.	3.37.	3.37.
Cryptographic Algorithm	3.4.	3.4.	3.4.
Cryptographic Domain Parameters	3.7.	3.7.	3.7.
Cryptographic Length	3.5.	3.5.	3.5.
Cryptographic Parameters	3.6.	3.6.	3.6.
Custom Attribute	3.33.	3.39.	3.39.
Deactivation Date	3.22.	3.27.	3.27.
Default Operation Policy	3.13.2.	3.18.2.	3.18.2.
Default Operation Policy for Certificates and Public Key Objects	3.13.2.2.	3.18.2.2.	3.18.2.2.
Default Operation Policy for Secret Objects	3.13.2.1.	3.18.2.1.	3.18.2.1.
Default Operation Policy for Template Objects	3.13.2.3.	3.18.2.3.	3.18.2.3.
Destroy Date	3.23.	3.28.	3.28.
Digest	3.12.	3.17.	3.17.
Digital Signature Algorithm	-	3.16.	3.16.
Fresh	-	3.34.	3.34.
Initial Date	3.18.	3.23.	3.23.
Key Value Location	-	-	3.42.
Key Value Present	-	-	3.41.
Last Change Date	3.32.	3.38.	3.38.
Lease Time	3.15.	3.20.	3.20.
Link	3.29.	3.35.	3.35.
Name	3.2.	3.2.	3.2.
Object Group	3.28.	3.33.	3.33.
Object Type	3.3.	3.3.	3.3.
Operation Policy Name	3.13.	3.18.	3.18.
Operations outside of operation policy control	3.13.1.	3.18.1.	3.18.1.
Original Creation Date	-	-	3.43.
Process Start Date	3.20.	3.25.	3.25.
Protect Stop Date	3.21.	3.26.	3.26.
Revocation Reason	3.26.	3.31.	3.31.
State	3.17.	3.22.	3.22.
Unique Identifier	3.1.	3.1.	3.1.
Usage Limits	3.16.	3.21.	3.21.
X.509 Certificate Identifier	-	3.10.	3.10.
X.509 Certificate Issuer	-	3.12.	3.12.
X.509 Certificate Subject	-	3.11.	3.11.
4 Client-to-Server Operations			
Activate	4.18.	4.19.	4.19.
Add Attribute	4.13.	4.14.	4.14.
Archive	4.21.	4.22.	4.22.
Cancel	4.25.	4.27.	4.27.
Certify	4.6.	4.7.	4.7.
Check	4.9.	4.10.	4.10.
Create	4.1.	4.1.	4.1.
Create Key Pair	4.2.	4.2.	4.2.

Reference Term	KMIP 1.0	KMIP 1.1	KMIP 1.2
Create Split Key	-	-	4.38.
Decrypt	-	-	4.30.
Delete Attribute	4.15.	4.16.	4.16.
Derive Key	4.5.	4.6.	4.6.
Destroy	4.20.	4.21.	4.21.
Discover Versions	-	4.26.	4.26.
Encrypt	-	-	4.29.
Get	4.10.	4.11.	4.11.
Get Attribute List	4.12.	4.13.	4.13.
Get Attributes	4.11.	4.12.	4.12.
Get Usage Allocation	4.17.	4.18.	4.18.
Hash	-	-	4.37.
Join Split Key	-	-	4.39.
Locate	4.8.	4.9.	4.9.
MAC	-	-	4.33.
MAC Verify	-	-	4.34.
Modify Attribute	4.14.	4.15.	4.15.
Obtain Lease	4.16.	4.17.	4.17.
Poll	4.26.	4.28.	4.28.
Query	4.24.	4.25.	4.25.
Re-certify	4.7.	4.8.	4.8.
Recover	4.22.	4.23.	4.23.
Register	4.3.	4.3.	4.3.
Re-key	4.4.	4.4.	4.4.
Re-key Key Pair	-	4.5.	4.5.
Revoke	4.19.	4.20.	4.20.
RNG Retrieve	-	-	4.35.
RNG Seed	-	-	4.36.
Sign	-	-	4.31.
Signature Verify	-	-	4.32.
Validate	4.23.	4.24.	4.24.
5 Server-to-Client Operations			
Notify	5.1.	5.1.	5.1.
Put	5.2.	5.2.	5.2.
6 Message Contents			
Asynchronous Correlation Value	6.8.	6.8.	6.8.
Asynchronous Indicator	6.7.	6.7.	6.7.
Attestation Capable Indicator	-	-	6.17.
Batch Count	6.14.	6.14.	6.14.
Batch Error Continuation Option	6.13.	6.13.	6.13.
Batch Item	6.15.	6.15.	6.15.
Batch Order Option	6.12.	6.12.	6.12.
Maximum Response Size	6.3.	6.3.	6.3.
Message Extension	6.16.	6.16.	6.16.
Operation	6.2.	6.2.	6.2.
Protocol Version	6.1.	6.1.	6.1.
Result Message	6.11.	6.11.	6.11.
Result Reason	6.10.	6.10.	6.10.
Result Status	6.9.	6.9.	6.9.
Time Stamp	6.5.	6.5.	6.5.
Unique Batch Item ID	6.4.	6.4.	6.4.
7 Message Format			

Reference Term	KMIP 1.0	KMIP 1.1	KMIP 1.2
Message Structure	7.1.	7.1.	7.1.
Operations	7.2.	7.2.	7.2.
8 Authentication			
Authentication	8	8	8
9 Message Encoding			
Alternative Name Type Enumeration	-	-	9.1.3.2.34.
Attestation Type Enumeration	-	-	9.1.3.2.36.
Batch Error Continuation Option Enumeration	9.1.3.2.29.	9.1.3.2.30.	9.1.3.2.30.
Bit Masks	9.1.3.3.	9.1.3.3.	9.1.3.3.
Block Cipher Mode Enumeration	9.1.3.2.13.	9.1.3.2.14.	9.1.3.2.14.
Cancellation Result Enumeration	9.1.3.2.24.	9.1.3.2.25.	9.1.3.2.25.
Certificate Request Type Enumeration	9.1.3.2.21.	9.1.3.2.22.	9.1.3.2.22.
Certificate Type Enumeration	9.1.3.2.6.	9.1.3.2.6.	9.1.3.2.6.
Credential Type Enumeration	9.1.3.2.1.	9.1.3.2.1.	9.1.3.2.1.
Cryptographic Algorithm Enumeration	9.1.3.2.12.	9.1.3.2.13.	9.1.3.2.13.
Cryptographic Usage Mask	9.1.3.3.1.	9.1.3.3.1.	9.1.3.3.1.
Defined Values	9.1.3.	9.1.3.	9.1.3.
Derivation Method Enumeration	9.1.3.2.20.	9.1.3.2.21.	9.1.3.2.21.
Digital Signature Algorithm Enumeration	-	9.1.3.2.7.	9.1.3.2.7.
Encoding Option Enumeration	-	9.1.3.2.32.	9.1.3.2.32.
Enumerations	9.1.3.2.	9.1.3.2.	9.1.3.2.
Examples	9.1.2.	9.1.2.	9.1.2.
Hashing Algorithm Enumeration	9.1.3.2.15.	9.1.3.2.16.	9.1.3.2.16.
Item Length	9.1.1.3.	9.1.1.3.	9.1.1.3.
Item Tag	9.1.1.1.	9.1.1.1.	9.1.1.1.
Item Type	9.1.1.2.	9.1.1.2.	9.1.1.2.
Item Value	9.1.1.4.	9.1.1.4.	9.1.1.4.
Key Compression Type Enumeration	9.1.3.2.2.	9.1.3.2.2.	9.1.3.2.2.
Key Format Type Enumeration	9.1.3.2.3.	9.1.3.2.3.	9.1.3.2.3.
Key Role Type Enumeration	9.1.3.2.16.	9.1.3.2.17.	9.1.3.2.17.
Key Value Location Type Enumeration	-	-	9.1.3.2.35.
Link Type Enumeration	9.1.3.2.19.	9.1.3.2.20.	9.1.3.2.20.
Name Type Enumeration	9.1.3.2.10.	9.1.3.2.11.	9.1.3.2.11.
Object Group Member Enumeration	-	9.1.3.2.33.	9.1.3.2.33.
Object Type Enumeration	9.1.3.2.11.	9.1.3.2.12.	9.1.3.2.12.
Opaque Data Type Enumeration	9.1.3.2.9.	9.1.3.2.10.	9.1.3.2.10.
Operation Enumeration	9.1.3.2.26.	9.1.3.2.27.	9.1.3.2.27.
Padding Method Enumeration	9.1.3.2.14.	9.1.3.2.15.	9.1.3.2.15.
Put Function Enumeration	9.1.3.2.25.	9.1.3.2.26.	9.1.3.2.26.
Query Function Enumeration	9.1.3.2.23.	9.1.3.2.24.	9.1.3.2.24.
Recommended Curve Enumeration for ECDSA, ECDH, and ECMQV	9.1.3.2.5.	9.1.3.2.5.	9.1.3.2.5.
Result Reason Enumeration	9.1.3.2.28.	9.1.3.2.29.	9.1.3.2.29.
Result Status Enumeration	9.1.3.2.27.	9.1.3.2.28.	9.1.3.2.28.
Revocation Reason Code Enumeration	9.1.3.2.18.	9.1.3.2.19.	9.1.3.2.19.
Secret Data Type Enumeration	9.1.3.2.8.	9.1.3.2.9.	9.1.3.2.9.
Split Key Method Enumeration	9.1.3.2.7.	9.1.3.2.8.	9.1.3.2.8.
State Enumeration	9.1.3.2.17.	9.1.3.2.18.	9.1.3.2.18.
Storage Status Mask	9.1.3.3.2.	9.1.3.3.2.	9.1.3.3.2.
Tags	9.1.3.1.	9.1.3.1.	9.1.3.1.
TTLV Encoding	9.1.	9.1.	9.1.
TTLV Encoding Fields	9.1.1.	9.1.1.	9.1.1.
Usage Limits Unit Enumeration	9.1.3.2.30.	9.1.3.2.31.	9.1.3.2.31.

Reference Term	**KMIP 1.0**	**KMIP 1.1**	**KMIP 1.2**
Validity Indicator Enumeration	9.1.3.2.22.	9.1.3.2.23.	9.1.3.2.23.
Wrapping Method Enumeration	9.1.3.2.4.	9.1.3.2.4.	9.1.3.2.4.
XML Encoding	9.2.	-	-
10 Transport			
Transport	10	10	10
12 KMIP Server and Client Implementation Conformance			
Conformance clauses for a KMIP Server	12.1.	-	-
KMIP Client Implementation Conformance	-	12.2.	12.2.
KMIP Server Implementation Conformance	-	12.1.	12.1.

395

Appendix C. Revision History

Revision	Date	Editor	Changes Made
wd01	26-June-2013	Tim Hudson / Bob Lockhart	Updated conformance wording style. Updated test case style. Included test cases for 1.0, 1.1 and 1.2. Applied new OASIS template.
wd02	6-August-2013	Tim Hudson / Bob Lockhart	Updated to include Permitted Test Case Variations and updated Test Cases based on July 2013 Interop
wd03	10-August-2013	Tim Hudson	Updated Permitted Test Case Variations
wd03a	24-October-2013	Tim Hudson	Editorial update to include VendorIdentification in the list of allowed variations as per TC motion.
pr01update	11-June-2014	Tim Hudson	Updated following Public Review