KMIP Suite B Profile Version 1.0

Committee Specification Draft ~~01~~02 / Public Review Draft ~~01~~02

~~09 January~~19 June 2014

Specification URIs

This version:

http://docs.oasis-open.org/kmip/kmip-suite-b-profile/v1.0/csprd02/kmip-suite-b-profile-v1.0-csprd02.doc (Authoritative)
http://docs.oasis-open.org/kmip/kmip-suite-b-profile/v1.0/csprd02/kmip-suite-b-profile-v1.0-csprd02.html
http://docs.oasis-open.org/kmip/kmip-suite-b-profile/v1.0/csprd02/kmip-suite-b-profile-v1.0-csprd02.pdf

Previous version:

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

~~**Previous version:**~~

~~N/A~~

Latest version:

http://docs.oasis-open.org/kmip/kmip-suite-b-profile/v1.0/kmip-suite-b-profile-v1.0.doc (Authoritative)
http://docs.oasis-open.org/kmip/kmip-suite-b-profile/v1.0/kmip-suite-b-profile-v1.0.html
http://docs.oasis-open.org/kmip/kmip-suite-b-profile/v1.0/kmip-suite-b-profile-v1.0.pdf

Technical Committee:

OASIS Key Management Interoperability Protocol (KMIP) TC

Chairs:

~~Robert Griffin (),~~ Subhash Sankuratripati (Subhash.Sankuratripati@netapp.com), NetApp
Saikat Saha (saikat.saha@oracle.com), Oracle

Editors:

Kelley Burgin (kwburgi@tycho.ncsc.mil), National Security Agency
Tim Hudson (tjh@cryptsoft.com), Cryptsoft

Related work:

This specification is related to:

- *Key Management Interoperability Protocol Profiles Version 1.0*. Edited by Robert Griffin and Subhash Sankuratripati. 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*. Edited by Robert Haas and Indra Fitzgerald. 24 January 2013. OASIS Standard. http://docs.oasis-open.org/kmip/spec/v1.1/os/kmip-spec-v1.1-os.html.

- *Key Management Interoperability Protocol Specification Version 1.2*. Edited by Kiran Thota and Kelley Burgin. Latest version~~.~~: http://docs.oasis-open.org/kmip/spec/v1.2/kmip-spec-v1.2.html.

Abstract:

Describes a profile for KMIP clients and KMIP servers using Suite B cryptography that has been approved by NIST for use by the U.S. Government and specified in NIST standards or recommendations.

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 https://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 (https://www.oasis-open.org/committees/kmip/ipr.php~~)~~.

Citation format:

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

[kmip-suite-b-v1.0]

KMIP Suite B Profile Version 1.0. Edited by Kelley Burgin and Tim Hudson. ~~09 January~~19 June 2014. OASIS Committee Specification Draft ~~01~~02 / Public Review Draft ~~01~~02. http://docs.oasis-open.org/kmip/kmip-suite-b-profile/v1.0/csprd02/kmip-suite-b-profile-v1.0-csprd02.html. Latest version: http://docs.oasis-open.org/kmip/kmip-suite-b-profile/v1.0/kmip-suite-b-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 https://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].~~

Suite B [SuiteB] requires that key establishment and signature algorithms be based upon Elliptic Curve Cryptography and that the encryption algorithm be AES [FIPS197]. Suite B includes:

Encryption	Advanced Encryption Standard (AES) (key sizes of 128 and 256 bits)
Digital Signature	Elliptic Curve Digital Signature Algorithm (ECDSA) (using the curves with 256-bit and 384-bit prime moduli)
Key Exchange	Elliptic Curve Diffie-Hellman (ECDH), (using the curves with 256-bit and 384-bit prime moduli)
Hashes	SHA-256 and SHA-384

Suite B provides for two levels of cryptographic security, namely a 128-bit minimum level of security (minLOS_128) and a 192-bit minimum level of security (minLOS_192). Each level defines a minimum strength that all cryptographic algorithms must provide. A KMIP product configured at a minimum level of security of 128 bits provides adequate protection for classified information up to the SECRET level. A KMIP product configured at a minimum level of security of 192 bits is required to protect classified information at the TOP SECRET level.

The Suite B non-signature primitives are divided into two columns as shown below.

	Column 1	Column 2
Encryption	AES-128	AES-256
Key Agreement	ECDH on P-256	ECDH on P-384
Hash for PRF/MAC	SHA-256	SHA-384

At the 128-bit minimum level of security, the non-signature primitives MUST either come exclusively from Column 1 or exclusively from Column 2.

At the 192-bit minimum level of security, the non-signature primitives MUST come exclusively from Column 2.

Digital signatures using ECDSA MUST be used for authentication. Following the direction of RFC 4754, ECDSA-256 represents an instantiation of the ECDSA algorithm using the P-256 curve and the SHA-256 hash function. ECDSA-384 represents an instantiation of the ECDSA algorithm using the P-384 curve and the SHA-384 hash function.

If configured at a minimum level of security of 128 bits, a KMIP product MUST use either ECDSA-256 or ECDSA-384 for authentication. It is allowable for one party to authenticate with ECDSA-256 and the other party to authenticate with ECDSA-384. This flexibility will allow interoperability between a KMIP client and server that have different sizes of ECDSA authentication keys. KMIP products configured at a minimum level of security of 128 bits MUST be able to verify ECDSA-256 signatures and SHOULD be able to verify

ECDSA-384 signatures. If configured at a minimum level of security of 192 bits, ECDSA-384 MUST be used by both the KMIP client and server for authentication. KMIP products configured at a minimum level of security of 192 bits MUST be able to verify ECDSA-384 signatures.

KMIP products, at both minimum levels of security, MUST each use an X.509 certificate that complies with the "Suite B Certificate and Certificate Revocation List (CRL) Profile" [RFC5759] and that contains an elliptic curve public key with the key usage bit set for digital signature.

1.1 Terminology

The key words “MUST”, “SHALL”, “SHOULD”, and “MAY” in this document are to be interpreted as described in [RFC2119].

1.2 Normative References

[CNSSP-15] N.S.A., “National Information Assurance Policy on the Use of Public Standards for the Secure Sharing of Information Among National Security Systems”, 1 October 2013, https://www.cnss.gov/Assets/pdf/CNSSP_No%2015_minorUpdate1_Oct12012.pdf.

[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.

[KMIP-ENCODE] *KMIP Additional Message Encodings Version 1.0.* URL Candidate OASIS Standard 01. DD MMM YYYY.

~~**[RFC4754]** D. Fu and J. Solinas, *IKE and IKEv2 Authentication Using the Elliptic Curve Digital Signature Algorithm (ECDSA)*, IETF RFC 4754, Jan 2007, .~~

[RFC5246] Dierks, T. and E. Rescorla, *The Transport Layer Security (TLS) Protocol Version 1.2*, IETF RFC 5246, August 2008, http://www.ietf.org/rfc/rfc5246.txt.

~~**[RFC6460]** M. Salter and R. Housley, *Suite B Profile for Transport Layer Security (TLS)*, IETF RFC 6460, January 2012, .~~

[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, 1 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, , Committee Note Draft, 1 December 2011.*~~

~~**[KMIP-UG-1_1]** *Key Management Interoperability Protocol Usage Guide Version 1.1*, , Committee Note Draft, 1 December 2011.~~

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

~~**[KMIP-TC-1_1]** *Key Management Interoperability Protocol Test Cases Version 1.1*, , Committee Note 01, 27 July 2012.~~

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

~~**[KMIP-UC]** *Key Management Interoperability Protocol Use Cases Version 1.0*, , Committee Specification, 15 June 2010.~~

[SuiteB] *Suite B Cryptography / Cryptographic Interoperability*, http://www.nsa.gov/ia/programs/suiteb_cryptography/

2 Suite B minLOS_128 Profile

The Suite B minLOS_128 Profile describes a KMIP client interacting with a KMIP server as an information assurance product to provide a minimum level of security of 128 bits.
(http://www.nsa.gov/ia/programs/suiteb_cryptography/)

2.1 Authentication Suite

Implementations conformant to this profile SHALL use TLS to negotiate a mutually-authenticated connection.

2.1.1 Protocols

Conformant KMIP clients and servers SHALL support:

- TLS v1.2 [RFC5246]

2.1.2 Cipher Suites

Conformant KMIP servers SHALL support the following cipher suites:

- TLS_ECDHE_ECDSA_WITH_AES_128_GCM_SHA256

2.1.3 Client Authenticity

Conformant KMIP servers and clients SHALL handle client authenticity in accordance with section 3.2.3 of the TLS 1.2 Authentication Suite [KMIP-PROF].

2.1.4 Object Owner

Conformant KMIP servers and clients SHALL handle object owner in accordance with section 3.2.4 of the TLS 1.2 Authentication Suite [KMIP-PROF].

2.1.5 KMIP Port Number

Conformant KMIP servers and clients SHALL handle the KMIP port number in in accordance with section 3.2.5 of the TLS 1.2 Authentication Suite [KMIP-PROF].

2.2 Suite B minLOS_128 - 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]:

~~1.~~2. SHALL conform to the *Baseline Client* ~~conformance clauses in~~ *Clause* (section 5.12) of [KMIP-PROF~~] and [KMIP-SPEC~~-1_1]

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

3. SHALL conform to the *Baseline Client* (section 5.2) of [KMIP-PROF-1_2]

KMIP clients conformant to this profile:

~~2.~~4. SHALL restrict use of the enumerated types listed in item 8~~6~~ of the server list ~~below~~in section 2.3 to the values noted against each item

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

6. MAY support extensions outside the scope of this standard (e.g., vendor extensions, conformance clauses) that do not conflict with any KMIP or [CNSSP-15] requirements.

2.3 Suite B minLOS_128 ~~MAY support any clause within [KMIP-SPEC] provided it does not conflict with any other clause within this section~~ - Server

~~1. .~~

~~2.1. MAY support extensions outside the scope of this standard (e.g., vendor extensions, conformance clauses) that do not conflict with any KMIP or [CNSSP-15] requirements.~~

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]:

~~3.~~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:

4. SHALL support the following *Objects* [KMIP-SPEC]
    a. *Certificate* [KMIP-SPEC]
    b. *Symmetric Key* [KMIP-SPEC]
    c. *Public Key* [KMIP-SPEC]
    d. *Private Key* [KMIP-SPEC]
5. SHALL support the following *Attributes* [KMIP-SPEC]
    a. *Cryptographic Algorithm* [KMIP-SPEC]
    b. *Cryptographic Length* [KMIP-SPEC] value :
        i. 128-bit (combined with AES)
        ii. 256-bit (combined with SHA, ECDH or ECDSA)
6. MAY support the following *Attributes* [KMIP-SPEC]
    a. *Cryptographic Length* [KMIP-SPEC] value :
        i. 256-bit (combined with AES)
        ii. 384-bit bit (combined with SHA, ECDH or ECDSA)
7. SHALL support the following C*lient-to-Server Operations* [KMIP-SPEC]:
    a. *Create* [KMIP-SPEC]
    b. *Create Key Pair* [KMIP-SPEC]
    c. *Register* [KMIP-SPEC]
    d. *Re-key* [KMIP-SPEC]
    e. *Re-key Key Pair* [KMIP-SPEC]
8. SHALL support the following *Message Encoding* [KMIP-SPEC]:
    a. *Recommended Curve Enumeration* [KMIP-SPEC] value:
        i. P-256 (SECP256R1)
    b. *Certificate Type Enumeration* [KMIP-SPEC] value:
        i. X.509
    c. *Cryptographic Algorithm Enumeration* [KMIP-SPEC] value:
        i. AES

ii. ECDSA
iii. ECDH
iv. HMAC-SHA256

d. *Hashing Algorithm Enumeration* [KMIP-SPEC]
i. SHA-256

e. *Object Type Enumeration* [KMIP-SPEC] value:
i. Certificate
ii. Symmetric Key
iii. Public Key
iv. Private Key

f. *Key Format Type Enumeration* [KMIP-SPEC] value:
i. Raw
ii. ECPrivateKey
iii. X.509
iv. Transparent ECDSA Private Key
v. Transparent ECDSA Public Key
vi. Transparent ECDH Private Key
vii. Transparent ECDH Public Key

g. *Digital Signature Algorithm Enumeration* [KMIP-SPEC] value:
i. ECDSA with SHA256 (on P-256)

9. MAY support the following *Message Encoding* [KMIP-SPEC]:
a. *Recommended Curve* [KMIP-SPEC] value:
i. P-384 (SECP384R1)

b. *Cryptographic Algorithm Enumeration* [KMIP-SPEC] value:
i. HMAC-SHA384

c. *Hashing Algorithm Enumeration* [KMIP-SPEC]
i. SHA-384

d. Digital Signature Algorithm Enumeration
i. ECDSA with SHA384 (on P-384)

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

11. MAY support extensions outside the scope of this standard (e.g., vendor extensions, conformance clauses) that do not conflict with any KMIP or [CNSSP-15] requirements.

~~3. SHALL support the returning results in accordance with the test cases.~~

~~4.1.~~ Suite B minLOS_128 ~~MAY support any clause within [KMIP-SPEC] provided it does not conflict with any other clause within this section 2.2.~~

~~5.1. MAY support extensions outside the scope of this standard (e.g., vendor extensions, conformance clauses) that do not conflict with any KMIP or [CNSSP-15] requirements.~~

3 ~~Suite B minLOS_128~~ Test Cases

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 or server system may vary as specified in section 6.10

3.1 Mandatory Suite B minLOS_128 Test Cases KMIP 1.0

~~This section documents the test cases that a client or server conformant to this profile SHALL support.~~

3.1.1 SUITEB_128-M-1-10 - Query

Perform a Query operation, querying the Operations and Objects supported by the server, and get a successful response.

The specific list of operations and object types returned in the response MAY vary.

The TLS protocol version and cipher suite SHALL be as specified in section 2.1

```
     # 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="Query"/>
0011     <RequestPayload>
0012       <QueryFunction type="Enumeration" value="QueryOperations"/>
0013       <QueryFunction type="Enumeration" value="QueryObjects"/>
0014     </RequestPayload>
0015   </BatchItem>
0016 </RequestMessage>
0017 <ResponseMessage>
0018   <ResponseHeader>
0019     <ProtocolVersion>
0020       <ProtocolVersionMajor type="Integer" value="1"/>
0021       <ProtocolVersionMinor type="Integer" value="0"/>
0022     </ProtocolVersion>
0023     <TimeStamp type="DateTime" value="2013-06-26T09:09:17+00:00"/>
0024     <BatchCount type="Integer" value="1"/>
0025   </ResponseHeader>
```

```
0026    <BatchItem>
0027      <Operation type="Enumeration" value="Query"/>
0028      <ResultStatus type="Enumeration" value="Success"/>
0029      <ResponsePayload>
0030        <Operation type="Enumeration" value="Query"/>
0031        <Operation type="Enumeration" value="Locate"/>
0032        <Operation type="Enumeration" value="Destroy"/>
0033        <Operation type="Enumeration" value="Get"/>
0034        <Operation type="Enumeration" value="Create"/>
0035        <Operation type="Enumeration" value="Register"/>
0036        <Operation type="Enumeration" value="GetAttributes"/>
0037        <Operation type="Enumeration" value="GetAttributeList"/>
0038        <Operation type="Enumeration" value="AddAttribute"/>
0039        <Operation type="Enumeration" value="ModifyAttribute"/>
0040        <Operation type="Enumeration" value="DeleteAttribute"/>
0041        <Operation type="Enumeration" value="Activate"/>
0042        <Operation type="Enumeration" value="Revoke"/>
0043        <Operation type="Enumeration" value="Poll"/>
0044        <Operation type="Enumeration" value="Cancel"/>
0045        <Operation type="Enumeration" value="Check"/>
0046        <Operation type="Enumeration" value="GetUsageAllocation"/>
0047        <Operation type="Enumeration" value="CreateKeyPair"/>
0048        <Operation type="Enumeration" value="ReKey"/>
0049        <Operation type="Enumeration" value="Archive"/>
0050        <Operation type="Enumeration" value="Recover"/>
0051        <Operation type="Enumeration" value="ObtainLease"/>
0052        <Operation type="Enumeration" value="Certify"/>
0053        <Operation type="Enumeration" value="ReCertify"/>
0054        <Operation type="Enumeration" value="Notify"/>
0055        <Operation type="Enumeration" value="Put"/>
0056        <ObjectType type="Enumeration" value="Certificate"/>
0057        <ObjectType type="Enumeration" value="SymmetricKey"/>
0058        <ObjectType type="Enumeration" value="SecretData"/>
0059        <ObjectType type="Enumeration" value="PublicKey"/>
0060        <ObjectType type="Enumeration" value="PrivateKey"/>
0061        <ObjectType type="Enumeration" value="Template"/>
0062        <ObjectType type="Enumeration" value="OpaqueObject"/>
0063        <ObjectType type="Enumeration" value="SplitKey"/>
0064      </ResponsePayload>
0065    </BatchItem>
0066  </ResponseMessage>
```

3.2 Mandatory Suite B minLOS_128 Test Cases KMIP 1.1

3.2.1 SUITEB_128-M-1-11 - Query

Perform a Query operation, querying the Operations and Objects supported by the server, and get a successful response.

The specific list of operations and object types returned in the response MAY vary.

The TLS protocol version and cipher suite SHALL be as specified in section 2.1

```
        # 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="Query"/>
0011      <RequestPayload>
0012        <QueryFunction type="Enumeration" value="QueryOperations"/>
0013        <QueryFunction type="Enumeration" value="QueryObjects"/>
0014      </RequestPayload>
0015    </BatchItem>
0016  </RequestMessage>
0017  <ResponseMessage>
0018    <ResponseHeader>
0019      <ProtocolVersion>
0020        <ProtocolVersionMajor type="Integer" value="1"/>
0021        <ProtocolVersionMinor type="Integer" value="1"/>
0022      </ProtocolVersion>
0023      <TimeStamp type="DateTime" value="2014-06-11T09:22:39+00:00"/>
0024      <BatchCount type="Integer" value="1"/>
0025    </ResponseHeader>
0026    <BatchItem>
0027      <Operation type="Enumeration" value="Query"/>
0028      <ResultStatus type="Enumeration" value="Success"/>
0029      <ResponsePayload>
0030        <Operation type="Enumeration" value="Query"/>
0031        <Operation type="Enumeration" value="Locate"/>
0032        <Operation type="Enumeration" value="Destroy"/>
0033        <Operation type="Enumeration" value="Get"/>
0034        <Operation type="Enumeration" value="Create"/>
0035        <Operation type="Enumeration" value="Register"/>
0036        <Operation type="Enumeration" value="GetAttributes"/>
0037        <Operation type="Enumeration" value="GetAttributeList"/>
0038        <Operation type="Enumeration" value="AddAttribute"/>
0039        <Operation type="Enumeration" value="ModifyAttribute"/>
0040        <Operation type="Enumeration" value="DeleteAttribute"/>
0041        <Operation type="Enumeration" value="Activate"/>
0042        <Operation type="Enumeration" value="Revoke"/>
0043        <Operation type="Enumeration" value="Poll"/>
0044        <Operation type="Enumeration" value="Cancel"/>
0045        <Operation type="Enumeration" value="Check"/>
0046        <Operation type="Enumeration" value="GetUsageAllocation"/>
0047        <Operation type="Enumeration" value="CreateKeyPair"/>
0048        <Operation type="Enumeration" value="ReKey"/>
0049        <Operation type="Enumeration" value="Archive"/>
0050        <Operation type="Enumeration" value="Recover"/>
0051        <Operation type="Enumeration" value="ObtainLease"/>
0052        <Operation type="Enumeration" value="ReKeyKeyPair"/>
0053        <Operation type="Enumeration" value="Certify"/>
0054        <Operation type="Enumeration" value="ReCertify"/>
0055        <Operation type="Enumeration" value="DiscoverVersions"/>
0056        <Operation type="Enumeration" value="Notify"/>
0057        <Operation type="Enumeration" value="Put"/>
0058        <ObjectType type="Enumeration" value="Certificate"/>
0059        <ObjectType type="Enumeration" value="SymmetricKey"/>
0060        <ObjectType type="Enumeration" value="SecretData"/>
0061        <ObjectType type="Enumeration" value="PublicKey"/>
0062        <ObjectType type="Enumeration" value="PrivateKey"/>
```

```
0063        <ObjectType type="Enumeration" value="Template"/>
0064        <ObjectType type="Enumeration" value="OpaqueObject"/>
0065        <ObjectType type="Enumeration" value="SplitKey"/>
0066      </ResponsePayload>
0067    </BatchItem>
0068  </ResponseMessage>
```

3.3 Mandatory Suite B minLOS_128 Test Cases KMIP 1.2

3.3.1 SUITEB_128-M-1-12 - Query

Perform a Query operation, querying the Operations and Objects supported by the server, and get a successful response.

The specific list of operations and object types returned in the response MAY vary.

The TLS protocol version and cipher suite SHALL be as specified in section 2.1~~N/A~~

```
      # 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="Query"/>
0011      <RequestPayload>
0012        <QueryFunction type="Enumeration" value="QueryOperations"/>
0013        <QueryFunction type="Enumeration" value="QueryObjects"/>
0014      </RequestPayload>
0015    </BatchItem>
0016  </RequestMessage>
```

```
0017  <ResponseMessage>
0018    <ResponseHeader>
0019      <ProtocolVersion>
0020        <ProtocolVersionMajor type="Integer" value="1"/>
0021        <ProtocolVersionMinor type="Integer" value="2"/>
0022      </ProtocolVersion>
0023      <TimeStamp type="DateTime" value="2014-06-11T09:23:21+00:00"/>
0024      <BatchCount type="Integer" value="1"/>
0025    </ResponseHeader>
0026    <BatchItem>
0027      <Operation type="Enumeration" value="Query"/>
0028      <ResultStatus type="Enumeration" value="Success"/>
0029      <ResponsePayload>
0030        <Operation type="Enumeration" value="Query"/>
0031        <Operation type="Enumeration" value="Locate"/>
0032        <Operation type="Enumeration" value="Destroy"/>
0033        <Operation type="Enumeration" value="Get"/>
0034        <Operation type="Enumeration" value="Create"/>
0035        <Operation type="Enumeration" value="Register"/>
0036        <Operation type="Enumeration" value="GetAttributes"/>
0037        <Operation type="Enumeration" value="GetAttributeList"/>
0038        <Operation type="Enumeration" value="AddAttribute"/>
0039        <Operation type="Enumeration" value="ModifyAttribute"/>
```

```
0040        <Operation type="Enumeration" value="DeleteAttribute"/>
0041        <Operation type="Enumeration" value="Activate"/>
0042        <Operation type="Enumeration" value="Revoke"/>
0043        <Operation type="Enumeration" value="Poll"/>
0044        <Operation type="Enumeration" value="Cancel"/>
0045        <Operation type="Enumeration" value="Check"/>
0046        <Operation type="Enumeration" value="GetUsageAllocation"/>
0047        <Operation type="Enumeration" value="CreateKeyPair"/>
0048        <Operation type="Enumeration" value="ReKey"/>
0049        <Operation type="Enumeration" value="Archive"/>
0050        <Operation type="Enumeration" value="Recover"/>
0051        <Operation type="Enumeration" value="ObtainLease"/>
0052        <Operation type="Enumeration" value="ReKeyKeyPair"/>
0053        <Operation type="Enumeration" value="Certify"/>
0054        <Operation type="Enumeration" value="ReCertify"/>
0055        <Operation type="Enumeration" value="DiscoverVersions"/>
0056        <Operation type="Enumeration" value="Notify"/>
0057        <Operation type="Enumeration" value="Put"/>
0058        <Operation type="Enumeration" value="RNGRetrieve"/>
0059        <Operation type="Enumeration" value="RNGSeed"/>
0060        <Operation type="Enumeration" value="Encrypt"/>
0061        <Operation type="Enumeration" value="Decrypt"/>
0062        <Operation type="Enumeration" value="Sign"/>
0063        <Operation type="Enumeration" value="SignatureVerify"/>
0064        <Operation type="Enumeration" value="MAC"/>
0065        <Operation type="Enumeration" value="MACVerify"/>
0066        <Operation type="Enumeration" value="Hash"/>
0067        <Operation type="Enumeration" value="CreateSplitKey"/>
0068        <Operation type="Enumeration" value="JoinSplitKey"/>
0069        <ObjectType type="Enumeration" value="Certificate"/>
0070        <ObjectType type="Enumeration" value="SymmetricKey"/>
0071        <ObjectType type="Enumeration" value="SecretData"/>
0072        <ObjectType type="Enumeration" value="PublicKey"/>
0073        <ObjectType type="Enumeration" value="PrivateKey"/>
0074        <ObjectType type="Enumeration" value="Template"/>
0075        <ObjectType type="Enumeration" value="OpaqueObject"/>
0076        <ObjectType type="Enumeration" value="SplitKey"/>
0077        <ObjectType type="Enumeration" value="PGPKey"/>
0078     </ResponsePayload>
0079   </BatchItem>
0080 </ResponseMessage>
```

242

4 Suite B minLOS_192 Profile

The Suite B minLOS_192 Profile describes a KMIP client interacting with a KMIP server as an information assurance product to provide a minimum level of security of 192 bits.
(http://www.nsa.gov/ia/programs/suiteb_cryptography/)

4.1 Authentication Suite

Implementations conformant to this profile SHALL use TLS to negotiate a mutually-authenticated connection.

4.1.1 Protocols

Conformant KMIP clients and servers SHALL support:

- TLS v1.2 [RFC5246]

4.1.2 Cipher Suites

Conformant KMIP servers SHALL support the following cipher suites:

- TLS_ECDHE_ECDSA_WITH_AES_256_GCM_SHA384

4.1.3 Client Authenticity

Conformant KMIP servers and clients SHALL handle client authenticity in accordance with section 3.2.3 of the TLS 1.2 Authentication Suite [KMIP-PROF].

4.1.4 Object Owner

Conformant KMIP servers and clients SHALL handle object owner in accordance with section 3.2.4 of the TLS 1.2 Authentication Suite [KMIP-PROF].

4.1.5 KMIP Port Number

Conformant KMIP servers and clients SHALL handle the KMIP port number in in accordance with section 3.2.5 of the TLS 1.2 Authentication Suite [KMIP-PROF].

4.2 Suite B minLOS_192 - 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]:

~~1.~~2. SHALL conform to the *Baseline Client* ~~conformance clauses in~~ *Clause* (section 5.12) of [KMIP-PROF~~] and [KMIP-SPEC~~-1_1]

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

3. SHALL conform to the *Baseline Client* (section 5.2) of [KMIP-PROF-1_2]

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

~~2.~~4. SHALL restrict use of the enumerated types listed in item 7~~5~~ of the server list ~~below~~in section 4.3 to the values noted against each item

~~3.~~5. MAY support any clause within [KMIP-SPEC] provided it does not conflict with any other clause within this section 4.2.

4.6. MAY support extensions outside the scope of this standard (e.g., vendor extensions, conformance clauses) that do not conflict with any KMIP or [CNSSP-15] requirements.

4.3 Suite B minLOS_192 - 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 under [KMIP-SPEC]:

2.4. SHALL support the following *Objects* [KMIP-SPEC]

a. *Certificate* [KMIP-SPEC]
b. *Symmetric Key* [KMIP-SPEC]
c. *Public Key* [KMIP-SPEC]
d. *Private Key* [KMIP-SPEC]

3.5. SHALL support the following *Attributes* [KMIP-SPEC]

e. *Cryptographic Algorithm* [KMIP-SPEC]
f. *Cryptographic Length* [KMIP-SPEC] value:
    i. 384-bit bit (combined with SHA, ECDH or ECDSA)

4.6. SHALL support the following C*lient-to-Server Operations* [KMIP-SPEC]:

g. *Create* [KMIP-SPEC]
h. *Create Key Pair* [KMIP-SPEC]
i. *Register* [KMIP-SPEC]
j. *Re-key* [KMIP-SPEC]
k. *Re-key Key Pair* [KMIP-SPEC]

5.7. SHALL support the following *Message Encoding* [KMIP-SPEC]:

l. *Recommended Curve Enumeration* [KMIP-SPEC] value:
    i. P-384 (SECP384R1)
m. *Certificate Type Enumeration* [KMIP-SPEC] value:
    i. X.509
n. *Cryptographic Algorithm Enumeration* [KMIP-SPEC] value:
    i. AES
    ii. ECDSA
    iii. ECDH
    iv. HMAC-SHA384
o. *Hashing Algorithm Enumeration* [KMIP-SPEC]
    i. SHA-384
p. *Object Type Enumeration* [KMIP-SPEC] value:
    i. Certificate
    ii. Symmetric Key
    iii. Public Key
    iv. Private Key

q. *Key Format Type Enumeration* [KMIP-SPEC] value:
    i. Raw
    ii. ECPrivateKey
    iii. X.509
    iv. Transparent ECDSA Private Key
    v. Transparent ECDSA Public Key
    vi. Transparent ECDH Private Key
    vii. Transparent ECDH Public Key
r. *Digital Signature Algorithm Enumeration* [KMIP-SPEC] value:
    i. ECDSA with SHA384 (on P-384)

~~2. SHALL support the returning results in accordance with the test cases.~~

~~6.~~8. MAY support any clause within [KMIP-SPEC] provided it does not conflict with any other clause within this section 4.3.

~~7.~~9. MAY support extensions outside the scope of this standard (e.g., vendor extensions, conformance clauses) that do not conflict with any KMIP or [CNSSP-15] requirements.

5 Suite B minLOS_192 Test Cases

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 or server system may vary as specified in section 6.10

5.1 Mandatory Suite B minLOS_192 Test Cases - KMIP v1.0

This section documents the test cases that a client or server conformant to this profile SHALL support.

5.1.1 ~~This section documents~~SUITEB_192-M-1-10 - Query

Perform a Query operation, querying the ~~test cases that a client or~~Operations and Objects supported by the server~~ conformant to this profile~~, and get a successful response.

The specific list of operations and object types returned in the response MAY vary.

The TLS protocol version and cipher suite SHALL ~~support.~~be as specified in section 4.1

```
      # 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="Query"/>
0011      <RequestPayload>
0012        <QueryFunction type="Enumeration" value="QueryOperations"/>
0013        <QueryFunction type="Enumeration" value="QueryObjects"/>
0014      </RequestPayload>
0015    </BatchItem>
0016  </RequestMessage>
0017  <ResponseMessage>
0018    <ResponseHeader>
0019      <ProtocolVersion>
0020        <ProtocolVersionMajor type="Integer" value="1"/>
0021        <ProtocolVersionMinor type="Integer" value="0"/>
0022      </ProtocolVersion>
0023      <TimeStamp type="DateTime" value="2013-06-26T09:09:17+00:00"/>
0024      <BatchCount type="Integer" value="1"/>
0025    </ResponseHeader>
```

```
0026    <BatchItem>
0027      <Operation type="Enumeration" value="Query"/>
0028      <ResultStatus type="Enumeration" value="Success"/>
0029      <ResponsePayload>
0030        <Operation type="Enumeration" value="Query"/>
0031        <Operation type="Enumeration" value="Locate"/>
0032        <Operation type="Enumeration" value="Destroy"/>
0033        <Operation type="Enumeration" value="Get"/>
0034        <Operation type="Enumeration" value="Create"/>
0035        <Operation type="Enumeration" value="Register"/>
0036        <Operation type="Enumeration" value="GetAttributes"/>
0037        <Operation type="Enumeration" value="GetAttributeList"/>
0038        <Operation type="Enumeration" value="AddAttribute"/>
0039        <Operation type="Enumeration" value="ModifyAttribute"/>
0040        <Operation type="Enumeration" value="DeleteAttribute"/>
0041        <Operation type="Enumeration" value="Activate"/>
0042        <Operation type="Enumeration" value="Revoke"/>
0043        <Operation type="Enumeration" value="Poll"/>
0044        <Operation type="Enumeration" value="Cancel"/>
0045        <Operation type="Enumeration" value="Check"/>
0046        <Operation type="Enumeration" value="GetUsageAllocation"/>
0047        <Operation type="Enumeration" value="CreateKeyPair"/>
0048        <Operation type="Enumeration" value="ReKey"/>
0049        <Operation type="Enumeration" value="Archive"/>
0050        <Operation type="Enumeration" value="Recover"/>
0051        <Operation type="Enumeration" value="ObtainLease"/>
0052        <Operation type="Enumeration" value="Certify"/>
0053        <Operation type="Enumeration" value="ReCertify"/>
0054        <Operation type="Enumeration" value="Notify"/>
0055        <Operation type="Enumeration" value="Put"/>
0056        <ObjectType type="Enumeration" value="Certificate"/>
0057        <ObjectType type="Enumeration" value="SymmetricKey"/>
0058        <ObjectType type="Enumeration" value="SecretData"/>
0059        <ObjectType type="Enumeration" value="PublicKey"/>
0060        <ObjectType type="Enumeration" value="PrivateKey"/>
0061        <ObjectType type="Enumeration" value="Template"/>
0062        <ObjectType type="Enumeration" value="OpaqueObject"/>
0063        <ObjectType type="Enumeration" value="SplitKey"/>
0064      </ResponsePayload>
0065    </BatchItem>
0066  </ResponseMessage>
```

356

357 ## 5.2 Mandatory Suite B minLOS_192 Test Cases KMIP 1.1

358 ### 5.2.1 SUITEB_192-M-1-11 - Query

359 Perform a Query operation, querying the Operations and Objects supported by the server, and get a
360 successful response.

361 The specific list of operations and object types returned in the response MAY vary.

362 The TLS protocol version and cipher suite SHALL be as specified in section 4.1

```
        # TIME 0
0001    <RequestMessage>
0002      <RequestHeader>
0003        <ProtocolVersion>
0004          <ProtocolVersionMajor type="Integer" value="1"/>
```

```
      <ProtocolVersionMinor type="Integer" value="1"/>
    </ProtocolVersion>
    <BatchCount type="Integer" value="1"/>
  </RequestHeader>
  <BatchItem>
    <Operation type="Enumeration" value="Query"/>
    <RequestPayload>
      <QueryFunction type="Enumeration" value="QueryOperations"/>
      <QueryFunction type="Enumeration" value="QueryObjects"/>
    </RequestPayload>
  </BatchItem>
</RequestMessage>
<ResponseMessage>
  <ResponseHeader>
    <ProtocolVersion>
      <ProtocolVersionMajor type="Integer" value="1"/>
      <ProtocolVersionMinor type="Integer" value="1"/>
    </ProtocolVersion>
    <TimeStamp type="DateTime" value="2014-06-11T09:22:39+00:00"/>
    <BatchCount type="Integer" value="1"/>
  </ResponseHeader>
  <BatchItem>
    <Operation type="Enumeration" value="Query"/>
    <ResultStatus type="Enumeration" value="Success"/>
    <ResponsePayload>
      <Operation type="Enumeration" value="Query"/>
      <Operation type="Enumeration" value="Locate"/>
      <Operation type="Enumeration" value="Destroy"/>
      <Operation type="Enumeration" value="Get"/>
      <Operation type="Enumeration" value="Create"/>
      <Operation type="Enumeration" value="Register"/>
      <Operation type="Enumeration" value="GetAttributes"/>
      <Operation type="Enumeration" value="GetAttributeList"/>
      <Operation type="Enumeration" value="AddAttribute"/>
      <Operation type="Enumeration" value="ModifyAttribute"/>
      <Operation type="Enumeration" value="DeleteAttribute"/>
      <Operation type="Enumeration" value="Activate"/>
      <Operation type="Enumeration" value="Revoke"/>
      <Operation type="Enumeration" value="Poll"/>
      <Operation type="Enumeration" value="Cancel"/>
      <Operation type="Enumeration" value="Check"/>
      <Operation type="Enumeration" value="GetUsageAllocation"/>
      <Operation type="Enumeration" value="CreateKeyPair"/>
      <Operation type="Enumeration" value="ReKey"/>
      <Operation type="Enumeration" value="Archive"/>
      <Operation type="Enumeration" value="Recover"/>
      <Operation type="Enumeration" value="ObtainLease"/>
      <Operation type="Enumeration" value="ReKeyKeyPair"/>
      <Operation type="Enumeration" value="Certify"/>
      <Operation type="Enumeration" value="ReCertify"/>
      <Operation type="Enumeration" value="DiscoverVersions"/>
      <Operation type="Enumeration" value="Notify"/>
      <Operation type="Enumeration" value="Put"/>
      <ObjectType type="Enumeration" value="Certificate"/>
      <ObjectType type="Enumeration" value="SymmetricKey"/>
      <ObjectType type="Enumeration" value="SecretData"/>
      <ObjectType type="Enumeration" value="PublicKey"/>
      <ObjectType type="Enumeration" value="PrivateKey"/>
```

```
0063        <ObjectType type="Enumeration" value="Template"/>
0064        <ObjectType type="Enumeration" value="OpaqueObject"/>
0065        <ObjectType type="Enumeration" value="SplitKey"/>
0066      </ResponsePayload>
0067    </BatchItem>
0068  </ResponseMessage>
```

5.3 Mandatory Suite B minLOS_192 Test Cases KMIP 1.2

5.3.1 SUITEB_192-M-1-12 - Query

Perform a Query operation, querying the Operations and Objects supported by the server, and get a successful response.

The specific list of operations and object types returned in the response MAY vary.

The TLS protocol version and cipher suite SHALL be as specified in section 4.1

~~N/A~~

```
      # 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="Query"/>
0011      <RequestPayload>
0012        <QueryFunction type="Enumeration" value="QueryOperations"/>
0013        <QueryFunction type="Enumeration" value="QueryObjects"/>
0014      </RequestPayload>
0015    </BatchItem>
0016  </RequestMessage>
```

```
0017  <ResponseMessage>
0018    <ResponseHeader>
0019      <ProtocolVersion>
0020        <ProtocolVersionMajor type="Integer" value="1"/>
0021        <ProtocolVersionMinor type="Integer" value="2"/>
0022      </ProtocolVersion>
0023      <TimeStamp type="DateTime" value="2014-06-11T09:23:21+00:00"/>
0024      <BatchCount type="Integer" value="1"/>
0025    </ResponseHeader>
0026    <BatchItem>
0027      <Operation type="Enumeration" value="Query"/>
0028      <ResultStatus type="Enumeration" value="Success"/>
0029      <ResponsePayload>
0030        <Operation type="Enumeration" value="Query"/>
0031        <Operation type="Enumeration" value="Locate"/>
0032        <Operation type="Enumeration" value="Destroy"/>
0033        <Operation type="Enumeration" value="Get"/>
0034        <Operation type="Enumeration" value="Create"/>
0035        <Operation type="Enumeration" value="Register"/>
0036        <Operation type="Enumeration" value="GetAttributes"/>
0037        <Operation type="Enumeration" value="GetAttributeList"/>
```

```
0038        <Operation type="Enumeration" value="AddAttribute"/>
0039        <Operation type="Enumeration" value="ModifyAttribute"/>
0040        <Operation type="Enumeration" value="DeleteAttribute"/>
0041        <Operation type="Enumeration" value="Activate"/>
0042        <Operation type="Enumeration" value="Revoke"/>
0043        <Operation type="Enumeration" value="Poll"/>
0044        <Operation type="Enumeration" value="Cancel"/>
0045        <Operation type="Enumeration" value="Check"/>
0046        <Operation type="Enumeration" value="GetUsageAllocation"/>
0047        <Operation type="Enumeration" value="CreateKeyPair"/>
0048        <Operation type="Enumeration" value="ReKey"/>
0049        <Operation type="Enumeration" value="Archive"/>
0050        <Operation type="Enumeration" value="Recover"/>
0051        <Operation type="Enumeration" value="ObtainLease"/>
0052        <Operation type="Enumeration" value="ReKeyKeyPair"/>
0053        <Operation type="Enumeration" value="Certify"/>
0054        <Operation type="Enumeration" value="ReCertify"/>
0055        <Operation type="Enumeration" value="DiscoverVersions"/>
0056        <Operation type="Enumeration" value="Notify"/>
0057        <Operation type="Enumeration" value="Put"/>
0058        <Operation type="Enumeration" value="RNGRetrieve"/>
0059        <Operation type="Enumeration" value="RNGSeed"/>
0060        <Operation type="Enumeration" value="Encrypt"/>
0061        <Operation type="Enumeration" value="Decrypt"/>
0062        <Operation type="Enumeration" value="Sign"/>
0063        <Operation type="Enumeration" value="SignatureVerify"/>
0064        <Operation type="Enumeration" value="MAC"/>
0065        <Operation type="Enumeration" value="MACVerify"/>
0066        <Operation type="Enumeration" value="Hash"/>
0067        <Operation type="Enumeration" value="CreateSplitKey"/>
0068        <Operation type="Enumeration" value="JoinSplitKey"/>
0069        <ObjectType type="Enumeration" value="Certificate"/>
0070        <ObjectType type="Enumeration" value="SymmetricKey"/>
0071        <ObjectType type="Enumeration" value="SecretData"/>
0072        <ObjectType type="Enumeration" value="PublicKey"/>
0073        <ObjectType type="Enumeration" value="PrivateKey"/>
0074        <ObjectType type="Enumeration" value="Template"/>
0075        <ObjectType type="Enumeration" value="OpaqueObject"/>
0076        <ObjectType type="Enumeration" value="SplitKey"/>
0077        <ObjectType type="Enumeration" value="PGPKey"/>
0078    </ResponsePayload>
0079  </BatchItem>
0080 </ResponseMessage>
```

371

6 Conformance

6.1 Suite B minLOS_128 Client KMIP V1.0 Profile Conformance

KMIP client implementations conformant to this profile:

1. SHALL support the Authentication Suite conditions as specified in Section 2.1 of this profile.
2. SHALL support the conditions as specified in Section 2.2 of this profile.
3. SHALL support all the Mandatory Suite B minLOS_128 Test Cases KMIP 1.0 (3.1~~and server~~)

6.2 Suite B minLOS_128 Client KMIP V1.1 Profile Conformance

KMIP client implementations conformant to this profile:

1. SHALL support the Authentication Suite conditions as specified in Section 2.1 of this profile.
2. SHALL support the conditions as specified in Section 2.2 of this profile.
3. SHALL support all the Mandatory Suite B minLOS_128 Test Cases KMIP 1.1 (3.2~~Suite B minLOS_192~~)

~~6.2~~6.3 Suite B minLOS_128 Client KMIP V1.2 Profile Conformance

KMIP client implementations conformant to this profile:

1. SHALL support the Authentication Suite conditions as specified in Section 2.1 of this profile.
2. SHALL support the conditions as specified in Section 2.2 of this profile.
3. SHALL support all the Mandatory Suite B minLOS_128 Test Cases KMIP 1.2 (3.3~~and~~)

6.4 Suite B minLOS_128 Server KMIP V1.0 Profile Conformance

KMIP server implementations conformant to this profile:

1. SHALL support the Authentication Suite conditions as specified in Section 2.1 of this profile.
2. SHALL support the conditions as specified in Section 2.3 of this profile.
3. SHALL support all the Mandatory Suite B minLOS_128 Test Cases KMIP 1.0 (3.1)

6.5 Suite B minLOS_128 Server KMIP V1.1 Profile Conformance

KMIP server implementations conformant to this profile:

1. SHALL support the Authentication Suite conditions as specified in Section 2.1 of this profile.
2. SHALL support the conditions as specified in Section 2.3 of this profile.
3. SHALL support all the Mandatory Suite B minLOS_128 Test Cases KMIP 1.1 (3.2)

6.6 Suite B minLOS_128 Server KMIP V1.2 Profile Conformance

KMIP server implementations conformant to this profile:

1. SHALL support the Authentication Suite conditions as specified in Section 2.1 of this profile.
2. SHALL support the conditions as specified in Section 2.3 of this profile.
3. SHALL support all the Mandatory Suite B minLOS_128 Test Cases KMIP 1.2 (3.3)

6.7 Suite B minLOS_192 Client KMIP V1.0 Profile Conformance

KMIP client implementations conformant to this profile:

1. SHALL support the Authentication Suite conditions as specified in Section 4.1 of this profile.
2. SHALL support the conditions as specified in Section 4.2 of this profile.
3. SHALL support all the Mandatory Suite B minLOS_192 Test Cases - KMIP v1.0 (5.1)

6.8 Suite B minLOS_192 Client KMIP V1.1 Profile Conformance

KMIP client implementations conformant to this profile:

1. SHALL support the Authentication Suite conditions as specified in Section 4.1 of this profile.
2. SHALL support the conditions as specified in Section 4.2 of this profile.
3. SHALL support all the Mandatory Suite B minLOS_192 Test Cases KMIP 1.1(5.2)

6.9 Suite B minLOS_192 Client KMIP V1.2 Profile Conformance

KMIP client implementations conformant to this profile:

1. SHALL support the Authentication Suite conditions as specified in Section 4.1 of this profile.
2. SHALL support the conditions as specified in Section 4.2 of this profile.
3. SHALL support all the Mandatory Suite B minLOS_192 Test Cases KMIP 1.2 (5.3)

6.10 Suite B minLOS_192 Server KMIP V1.0 Profile Conformance

KMIP server implementations conformant to this profile:

1. SHALL support the Authentication Suite conditions as specified in Section 4.1 of this profile.
2. SHALL support the conditions as specified in Section 4.3 of this profile.
3. SHALL support all the Mandatory Suite B minLOS_192 Test Cases - KMIP v1.0 (5.1)

6.11 Suite B minLOS_192 Server KMIP V1.1 Profile Conformance

KMIP server implementations conformant to this profile:

1. SHALL support the Authentication Suite conditions as specified in Section 4.1 of this profile.
2. SHALL support the conditions as specified in Section 4.3 of this profile.
3. SHALL support all the Mandatory Suite B minLOS_192 Test Cases KMIP 1.1(5.2)

6.12 Suite B minLOS_192 Server KMIP V1.2 Profile Conformance

KMIP server implementations conformant to this profile:

1. SHALL support the Authentication Suite conditions as specified in Section 4.1 of this profile.
2. SHALL support the conditions as specified in Section 4.3 of this profile.
3. SHALL support all the Mandatory Suite B minLOS_192 Test Cases KMIP 1.2 (5.3)

~~6.3~~6.13 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.

~~6.3.1~~6.13.1 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:

20. Object Group values – May or may not return one or more Object Group values not included in the requests
21. y-CustomAttributes – May or may not include additional server-specific associated attributes not included in requests
22. Message Extensions – May or may not include additional (non-critical) vendor extensions
23. TemplateAttribute – May or may not be included in responses where the Template Attribute response is noted as optional in [KMIP-SPEC]
24. AttributeIndex – May or may not include Attribute Index value where the Attribute Index value is 0 for Protocol Versions 1.1 and above.
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.
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.
27. VendorIdentification - The value (if included) may vary from the text contained within the test case.

~~6.3.2~~6.13.2 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)

623 Michael Yoder, Vormetric. Inc.
624 Magda Zdunkiewicz, Cryptsoft
625 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.

626

Appendix C. Revision History

Revision	Date	Editor	Changes Made
wd01	10 July 2013	Kelley Burgin / Tim Hudson	Initial Draft
wd02	8 August 2013	Kelley Burgin	Editorial updates and inclusion of a corresponding restriction on client enumeration usage
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

627