

Key Management Interoperability Protocol Profiles Version 1.1

OASIS Standard

24 January 2013

Specification URIs

This version:

<http://docs.oasis-open.org/kmip/profiles/v1.1/os/kmip-profiles-v1.1-os.doc> (Authoritative)
<http://docs.oasis-open.org/kmip/profiles/v1.1/os/kmip-profiles-v1.1-os.html>
<http://docs.oasis-open.org/kmip/profiles/v1.1/os/kmip-profiles-v1.1-os.pdf>

Previous version:

<http://www.oasis-open.org/committees/download.php/44884/kmip-profiles-v1.1-csprd01.zip>

Latest version:

<http://docs.oasis-open.org/kmip/profiles/v1.1/kmip-profiles-v1.1.doc> (Authoritative)
<http://docs.oasis-open.org/kmip/profiles/v1.1/kmip-profiles-v1.1.html>
<http://docs.oasis-open.org/kmip/profiles/v1.1/kmip-profiles-v1.1.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:

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

Related work:

This specification replaces or supersedes:

- *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>.

This specification is related to:

- *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 Test Cases Version 1.1*. Latest version. <http://docs.oasis-open.org/kmip/testcases/v1.1/kmip-testcases-v1.1.html>
- *Key Management Interoperability Protocol Usage Guide Version 1.1*. Latest version. <http://docs.oasis-open.org/kmip/ug/v1.1/kmip-ug-v1.1.html>

Abstract:

This document is intended for developers and architects who wish to design systems and applications that conform to the Key Management Interoperability Protocol specification.

KMIP V1.1 enhances the KMIP V1.0 standard (established in October 2010) by

- 1) defining new functionality in the protocol to improve interoperability, such as a Discover Versions operation and a Group object;
- 2) defining additional Test Cases for verifying and validating the new functionality;

- 3) providing additional information in the KMIP Usage Guide to assist in effective implementation of KMIP in key management clients and servers; and
- 4) defining new profiles for establishing KMIP-compliant implementations.

The Key Management Interoperability Protocol (KMIP) is a single, comprehensive protocol for communication between clients that request any of a wide range of encryption keys and servers that store and manage those keys. By replacing redundant, incompatible key management protocols, KMIP provides better data security while at the same time reducing expenditures on multiple products.

Status:

This document was last revised or approved by the membership of OASIS 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/>.

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>).

Citation format:

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

[KMIP-Profiles]

Key Management Interoperability Protocol Profiles Version 1.1. 24 January 2013. OASIS Standard. <http://docs.oasis-open.org/kmip/profiles/v1.1/os/kmip-profiles-v1.1-os.html>.

Notices

Copyright © OASIS Open 2013. All Rights Reserved.

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.....	7
1.1	Terminology.....	7
1.2	Normative References.....	7
1.3	Non-Normative References.....	7
2	Profiles.....	9
2.1	Guidelines for Specifying Conformance Clauses.....	9
2.2	Guidelines for Specifying Authentication Suites.....	9
2.3	Guidelines for Specifying KMIP Profiles.....	9
2.4	Guidelines for Validating Conformance to KMIP Profiles.....	9
3	Authentication Suites.....	11
3.1	Basic Authentication Suite.....	11
3.1.1	Protocols.....	11
3.1.2	Cipher Suites.....	11
3.1.3	Client Authenticity.....	11
3.1.4	Object Owner.....	11
3.1.5	KMIP Port Number.....	12
3.2	TLS 1.2 Authentication Suite.....	12
3.2.1	Protocols.....	12
3.2.2	Cipher Suites.....	12
3.2.3	Client Authenticity.....	12
3.2.4	Object Owner.....	12
3.2.5	KMIP Port Number.....	12
4	KMIP Profiles.....	13
4.1	Basic Discover Versions Server Profile.....	13
4.2	Basic Baseline Server KMIP Profile.....	13
4.3	Basic Secret Data Server KMIP Profile.....	13
4.4	Basic Symmetric Key Store and Server KMIP Profile.....	13
4.5	Basic Symmetric Key Foundry and Server KMIP Profile.....	13
4.6	Basic Asymmetric Key Store Server KMIP Profile.....	13
4.7	Basic Asymmetric Key and Certificate Store Server KMIP Profile.....	13
4.8	Basic Asymmetric Key Foundry and Server KMIP Profile.....	13
4.9	Basic Certificate Server KMIP Profile.....	13
4.10	Basic Asymmetric Key Foundry and Certificate Server KMIP Profile.....	13
4.11	Discover Versions TLS 1.2 Authentication Server Profile.....	14
4.12	Baseline Server TLS 1.2 Authentication KMIP Profile.....	14
4.13	Secret Data Server TLS 1.2 Authentication KMIP Profile.....	14
4.14	Symmetric Key Store and Server TLS 1.2 Authentication KMIP Profile.....	14
4.15	Symmetric Key Foundry and Server TLS 1.2 Authentication KMIP Profile.....	14
4.16	Asymmetric Key Store Server TLS 1.2 Authentication KMIP Profile.....	14
4.17	Asymmetric Key and Certificate Store Server TLS 1.2 Authentication KMIP Profile.....	14
4.18	Asymmetric Key Foundry and Server TLS 1.2 Authentication KMIP Profile.....	14
4.19	Certificate Server TLS 1.2 Authentication KMIP Profile.....	14
4.20	Asymmetric Key Foundry and Certificate Server TLS 1.2 Authentication KMIP Profile.....	14

4.21 Basic Discover Versions Client KMIP Profile.....	15
4.22 Basic Baseline Client KMIP Profile	15
4.23 Basic Secret Data Client KMIP Profile.....	15
4.24 Basic Symmetric Key Store Client KMIP Profile.....	15
4.25 Basic Symmetric Key Foundry Client KMIP Profile	15
4.26 Basic Asymmetric Key Store Client KMIP Profile	15
4.27 Basic Asymmetric Key and Certificate Store Client KMIP Profile.....	15
4.28 Basic Asymmetric Key Foundry Client KMIP Profile	15
4.29 Basic Certificate Client KMIP Profile	15
4.30 Basic Asymmetric Key Foundry and Certificate Client KMIP Profile	15
4.31 Discover Versions Client TLS 1.2 Authentication KMIP Profile.....	15
4.32 Baseline Client TLS 1.2 Authentication KMIP Profile	15
4.33 Secret Data Client TLS 1.2 Authentication KMIP Profile	16
4.34 Symmetric Key Store Client TLS 1.2 Authentication KMIP Profile	16
4.35 Symmetric Key Foundry Client TLS 1.2 Authentication KMIP Profile	16
4.36 Asymmetric Key Store Client TLS 1.2 Authentication KMIP Profile	16
4.37 Asymmetric Key and Certificate Store Client TLS 1.2 Authentication KMIP Profile.....	16
4.38 Asymmetric Key Foundry Client TLS 1.2 Authentication KMIP Profile	16
4.39 Certificate Client TLS 1.2 Authentication KMIP Profile.....	16
4.40 Asymmetric Key Foundry and Certificate Client TLS 1.2 Authentication KMIP Profile	16
4.41 Storage Client KMIP Profile	16
4.42 Storage Client TLS 1.2 Authentication KMIP Profile	16
5 Conformance Clauses.....	17
5.1 Discover Versions Server Clause	17
5.1.1. Implementation Conformance.....	17
5.1.2 Conformance of a Discover Versions Server	17
5.2 Baseline Server Clause	17
5.2.1 Implementation Conformance	17
5.2.2 Conformance of a KMIP Baseline Server	17
5.3 Secret Data Server Clause	18
5.3.1 Implementation Conformance	19
5.3.2 Conformance of a Secret Data Server	19
5.4 Symmetric Key Store and Server Conformance Clause	19
5.4.1 Implementation Conformance	19
5.4.2 Conformance as a Symmetric Key Store and Server	19
5.5 Symmetric Key Foundry and Server Conformance Clause.....	20
5.5.1 Implementation Conformance	20
5.5.2 Conformance as a KMIP Symmetric Key Foundry and Server	20
5.6 Asymmetric Key Store Server Conformance Clauses.....	21
5.6.1 Implementation Conformance	21
5.6.2 Conformance as an Asymmetric Key Store Server	21
5.7 Asymmetric Key and Certificate Store Server Conformance Clauses	22
5.7.1 Implementation Conformance	22
5.7.2 Conformance as a Asymmetric Key and Certificate Store Server	22
5.8 Asymmetric Key Foundry and Server Conformance Clauses	23

5.8.1 Implementation Conformance	23
5.8.2 Conformance as a Asymmetric Key Foundry and Server	23
5.9 Certificate Server Conformance Clauses	24
5.9.1 Implementation Conformance	24
5.9.2 Conformance as a Certificate Server	24
5.10 Asymmetric Key Foundry and Certificate Server Conformance Clauses.....	25
5.10.1 Implementation Conformance	25
5.10.2 Conformance as a Asymmetric Key Foundry and Certificate Server.....	25
5.11 Discover Versions Client Clause	26
5.11.1 Implementation Conformance	26
5.11.2 Conformance of a Discover Versions Client	26
5.12 Baseline Client Clause.....	26
5.12.1 Implementation Conformance	26
5.12.2 Conformance of a KMIP Baseline Client.....	26
5.13 Secret Data Client Clause	28
5.13.1 Implementation Conformance	28
5.13.2 Conformance of a Secret Data Client	28
5.14 Symmetric Key Store Client Conformance Clause.....	28
5.14.1 Implementation Conformance	28
5.14.2 Conformance as a Symmetric Key Store Client.....	29
5.15 Symmetric Key Foundry Client Conformance Clause	29
5.15.1 Implementation Conformance	29
5.15.2 Conformance as a KMIP Symmetric Key Foundry Client	29
5.16 Asymmetric Key Store Client Conformance Clauses	30
5.16.1 Implementation Conformance	30
5.16.2 Conformance as a Asymmetric Key Store Client.....	30
5.17 Asymmetric Key and Certificate Store Client Conformance Clauses.....	31
5.17.1 Implementation Conformance	31
5.17.2 Conformance as an Asymmetric Key and Certificate Store Client.....	31
5.18 Asymmetric Key Foundry Client Conformance Clauses	32
5.18.1 Implementation Conformance	32
5.18.2 Conformance as an Asymmetric Key Foundry Client	32
5.19 Certificate Client Conformance Clauses.....	33
5.19.1 Implementation Conformance	33
5.19.2 Conformance as a Basic Certificate Client.....	33
5.20 Asymmetric Key Foundry and Certificate Client Conformance Clauses	34
5.20.1 Implementation Conformance	34
5.20.2 Conformance as a Basic Asymmetric Key Foundry and Certificate Client	34
5.21 Storage Client Conformance Clauses	35
5.21.1 Implementation Conformance	35
5.21.2 Conformance as a Storage Client	35
A. Acknowledgements.....	37
B. Revision History.....	39

1 Introduction

OASIS requires a conformance section in an approved committee specification ([KMIP-Spec] [TCProc], section 2.18 Work Product Quality, paragraph 8a):

A specification that is approved by the TC at the Public Review Draft, Committee Specification or OASIS Standard level must include a separate section, listing a set of numbered conformance clauses, to which any implementation of the specification must adhere in order to claim conformance to the specification (or any optional portion thereof).

This document intends to meet this OASIS requirement on conformance clauses for a KMIP Server or KMIP Client ([KMIP-Spec] 12.1, 12.2) through profiles that define the use of KMIP objects, attributes, operations, message elements and authentication methods within specific contexts of KMIP server and client interaction. These profiles define a set of normative constraints for employing KMIP within a particular environment or context of use. They may, optionally, require the use of specific KMIP functionality or in other respects define the processing rules to be followed by profile actors.

For normative definition of the elements of KMIP specified in these profiles, see the KMIP Specification ([KMIP-Spec]). Illustrative guidance for the implementation of KMIP clients and servers is provided in the KMIP Usage Guide ([KMIP-UG]) and KMIP Test Cases ([KMIP_TC]).

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 .

1.2 Normative References

- [KMIP-Spec] *Key Management Interoperability Protocol Specification Version 1.1*. 21 September 2012. Candidate OASIS Standard 01.
<http://docs.oasis-open.org/kmip/spec/v1.1/cos01/kmip-spec-v1.1-cos01.html>.
- [RFC2119] S. Bradner, *Key words for use in RFCs to Indicate Requirement Levels*, <http://www.ietf.org/rfc/rfc2119.txt>, IETF RFC 2119, March 1997.
- [RFC 2246] T. Dierks & C.Allen, *The TLS Protocol, Version 1.0*, <http://www.ietf.org/rfc/rfc2246.txt>, IETF RFC 2246, January 1999
- [RFC 3268] P. Chown, *Advanced Encryption Standard (AES) Ciphersuites for Transport Layer Security (TLS)*, <http://www.ietf.org/rfc/rfc3268.txt>, IETF RFC 3268, June 2002
- [RFC 4346] T. Dierks & E. Rescorla, *The Transport Layer Security (TLS) Protocol, Version 1.1*, <http://www.ietf.org/rfc/rfc4346.txt>, IETF RFC 4346, April 2006
- [RFC 5246] T. Dierks & E. Rescorla, *The Transport Layer Security (TLS) Protocol, Version 1.2*, <http://www.ietf.org/rfc/rfc5246.txt>, IETF RFC 5246, August 2008
- [NIST 800-57 Part 3] Barker, Burr, et.al, *Recommendation for Key Management Part 3: Application-Specific Key Management Guidance*, http://csrc.nist.gov/publications/nistpubs/800-57/sp800-57_PART3_key-management_Dec2009.pdf, December 2009

1.3 Non-Normative References

- [KMIP-G] *Key Management Interoperability Protocol Usage Guide Version 1.1*. 27 July 2012. OASIS Committee Note 01.
<http://docs.oasis-open.org/kmip/ug/v1.1/cn01/kmip-ug-v1.1-cn01.html>.

45 **[KMTC]**

Key Management Interoperability Protocol Test Cases Version 1.1. 27 July 2012.
OASIS Committee Note 01. [http://docs.oasis-
open.org/kmip/testcases/v1.1/cn01/kmip-testcases-v1.1-cn01.html](http://docs.oasis-open.org/kmip/testcases/v1.1/cn01/kmip-testcases-v1.1-cn01.html).

46

47

48

49

50

51 2 Profiles

52 This document defines a selected set of conformance clauses and authentication suites which when
53 “paired together” form KMIP Profiles. The KMIP TC also welcomes proposals for new profiles. KMIP TC
54 members are encouraged to submit these proposals to the KMIP TC for consideration for inclusion in a
55 future version of this TC-approved document. However, some OASIS members may simply wish to inform
56 the committee of profiles or other work related to KMIP.

57 2.1 Guidelines for Specifying Conformance Clauses

58 This section provides a checklist of issues that SHALL be addressed by each clause.

- 59 1. Implement functionality as mandated by [KMIP-Spec] Section 12 (Conformance clauses for a
60 KMIP server or a KMIP client)
- 61 2. Specify the list of additional objects that SHALL be supported
- 62 3. Specify the list of additional attributes that SHALL be supported
- 63 4. Specify the list of additional operations that SHALL be supported
- 64 5. Specify any additional message content that SHALL be supported

65 2.2 Guidelines for Specifying Authentication Suites

- 66 1. Channel Security – For all operations, communication between Client and Server SHALL
67 establish and maintain channel confidentiality and integrity,.
- 68 2. Channel Options – Options like protocol version and cipher suite
- 69 3. Server and Client Authenticity – For all operations, communication between Client and Server
70 SHALL provide assurance of server authenticity and client authenticity

71 2.3 Guidelines for Specifying KMIP Profiles

72 A KMIP profile is a tuple of {Conformance Clause, Authentication Suite}.

73 Any vendor or organization, such as other standards bodies, MAY create a KMIP Profile and publish it.

- 74 • The profile SHALL be publicly available.
- 75 • The KMIP Technical Committee SHALL be formally advised of the availability of the profile and
76 the location of the published profile.
- 77 • The profile SHALL be defined as a tuple of {Conformance Clause, Authentication Suite}.

78 2.4 Guidelines for Validating Conformance to KMIP Profiles

79 A KMIP server implementation SHALL claim conformance to a specific server profile only if it
80 instruments all required objects, operations, messaging and attributes of that profile

- 81 • All objects specified as required in that profile
- 82 • All operations specified as required in that profile
- 83 • All attributes specified as required in that profile
- 84 • The defined wire protocols (TLS, SSL, IPSec, etc...) for that profile
- 85 • The defined methods of authentication for that profile

86 A KMIP client implementation SHALL claim conformance to a specific client profile only if it
87 instruments all required objects, operations, messaging and attributes of that profile

- 88 • All objects specified as required in that profile
- 89 • All operations specified as required in that profile
- 90 • All attributes specified as required in that profile
- 91 • The defined wire protocols (TLS, SSL, IPSec, etc...) for that profile
- 92 • The defined methods of authentication for that profile
- 93

94 3 Authentication Suites

95 This section contains the list of protocol versions and cipher suites that are to be used by profiles
96 contained within this document.

97 3.1 Basic Authentication Suite

98 This authentication set stipulates that a KMIP client and server SHALL use TLS to negotiate a mutually-
99 authenticated connection.

100 3.1.1 Protocols

101 Conformant KMIP servers SHALL support TLSv1.0. They MAY support TLS v1.1 [RFC 4346], TLS v1.2
102 [RFC 5246] bearing in mind that they are not compatible with each other and SHALL NOT support
103 SSLv3.0, SSLv2.0 and SSLv1.0.

104 3.1.2 Cipher Suites

105 Conformant KMIP servers SHALL support the following cipher suites:

- 106 • TLS_RSA_WITH_AES_128_CBC_SHA

107 Basic Authentication Suite Conformant KMIP servers MAY support the cipher suites listed in tables 4-1
108 through 4-4 of NIST 800-57 Part 3 with the exception of NULL ciphers (at the time this document was
109 created, the only NULL cipher in 800-57 Part 3 was: TLS_RSA_WITH_NONE_SHA)

110 Basic Authentication Suite Conformant KMIP servers SHALL NOT support any other cipher suites.

111 NOTE: TLS 1.0 has some security issues as described in <http://www.openssl.org/~bodo/tls-cbc.txt>.
112 Implementations that need protections against this attack should considering using the “TLS 1.2
113 Authentication Suite”

114 *At the time this document was published, NIST 800-57 Part 3 Table 4-1, for cipher suites that have both*
115 *SHA1 and SHA256 variants, erroneously categorizes SHA256 based ciphers under TLS versions 1.0, 1.1*
116 *and 1.2 and SHA1 based ciphers under TLS 1.2. The correct value for SHA256 based ciphers should 1.2*
117 *and for SHA1 based ciphers it should be 1.0, 1.1 and 1.2.*

118 3.1.3 Client Authenticity

119 For authenticated services KMIP servers SHALL require the use of channel (TLS) mutual authentication
120 to provide assurance of client authenticity.

121
122 In the absence of Credential information in the request header, KMIP servers SHALL use the identity
123 derived from the channel authentication as the client identity.

124
125 In the presence of Credential information in the request header, KMIP servers SHALL consider such
126 Credential information into their evaluation of client authenticity and identity, along with the authenticity
127 and identity derived from the channel. The exact mechanisms for such evaluation are outside the scope
128 of this specification.

129 3.1.4 Object Owner

130 KMIP objects have an `owner`. For those KMIP requests that result in new managed objects the client
131 identity SHALL be used as the owner of the managed object. For those operations that only access pre-
132 existent managed objects, the client identity SHALL be checked against the owner and access SHALL be
133 controlled as detailed in section 3.18 of [KMIP-SPEC].

134 **3.1.5 KMIP Port Number**

135 KMIP servers using the Basic Authentication Suite SHOULD use TCP port number 5696, as assigned by
136 IANA, to receive and send KMIP messages. KMIP clients using the Basic Authentication Suite MAY use
137 the same 5696 TCP port number.

138

139 **3.2 TLS 1.2 Authentication Suite**

140 This authentication set stipulates that a KMIP client and server SHALL use TLS to negotiate a mutually-
141 authenticated connection.

142 **3.2.1 Protocols**

143 Conformant KMIP servers SHALL support TLSv1.2

144 **3.2.2 Cipher Suites**

145 Conformant KMIP servers SHALL support the following cipher suites:

146

- TLS_RSA_WITH_AES_256_CBC_SHA256

147

- TLS_RSA_WITH_AES_128_CBC_SHA256

148 TLS 1.2 Authentication Suite Conformant KMIP servers MAY support the cipher suites listed in tables 4-1
149 through 4-4 of NIST 800-57 Part 3 with the exception of NULL ciphers (at the time this document was
150 created, the only NULL cipher in 800-57 Part 3 was: TLS_RSA_WITH_NONE_SHA)

151 TLS 1.2 Authentication Suite Conformant KMIP servers SHALL NOT support any other cipher suites

152 NIST 800-57 Part 3 Table 4-1, for cipher suites that have both SHA1 and SHA256 variants, erroneously
153 categorizes SHA256 based ciphers under TLS versions 1.0, 1.1 and 1.2 and SHA1 based ciphers under
154 TLS 1.2. The correct value for SHA256 based ciphers should be 1.2 and for SHA1 based ciphers it should
155 be 1.0, 1.1 and 1.2.

156 **3.2.3 Client Authenticity**

157 Same as the basic authentication suite Section 3.1.3.

158 **3.2.4 Object Owner**

159 Same as the basic authentication suite Section 3.1.4.

160 **3.2.5 KMIP Port Number**

161 Same as the basic authentication suite Section 3.1.5.

162 4 KMIP Profiles

163 This section lists the KMIP profiles that are defined in this specification. More than one profile may be
164 supported at the same time provided there are no conflicting requirements.

165 4.1 Basic Discover Versions Server Profile

166 A profile that consists of the tuple {Discover Versions Server Conformance Clause, Basic Authentication
167 Suite}

168 4.2 Basic Baseline Server KMIP Profile

169 A profile that consists of the tuple {Baseline Server Conformance Clause, Basic Authentication Suite}

170 4.3 Basic Secret Data Server KMIP Profile

171 A profile that consists of the tuple {Secret Data Server Conformance Clause, Basic Authentication Suite}

172 4.4 Basic Symmetric Key Store and Server KMIP Profile

173 A profile that consists of the tuple {Basic Symmetric Key Store and Server Conformance Clause, Basic
174 Authentication Suite}

175 4.5 Basic Symmetric Key Foundry and Server KMIP Profile

176 A profile that consists of the tuple {Basic Symmetric Key Foundry and Server Conformance Clause, Basic
177 Authentication Suite}

178 4.6 Basic Asymmetric Key Store Server KMIP Profile

179 A profile that consists of the tuple {Basic Asymmetric Key Store Server Conformance Clause, Basic
180 Authentication Suite}.

181 4.7 Basic Asymmetric Key and Certificate Store Server KMIP Profile

182 A profile that consists of the tuple {Basic Asymmetric Key and Certificate Store Server Conformance
183 Clause, Basic Authentication Suite}.

184 4.8 Basic Asymmetric Key Foundry and Server KMIP Profile

185 A profile that consists of the tuple {Basic Asymmetric Key Foundry and Server Conformance Clause,
186 Basic Authentication Suite}.

187 4.9 Basic Certificate Server KMIP Profile

188 A profile that consists of the tuple {Basic Certificate Server Conformance Clause, Basic Authentication
189 Suite}.

190 4.10 Basic Asymmetric Key Foundry and Certificate Server KMIP 191 Profile

192 A profile that consists of the tuple {Basic Asymmetric Key Foundry and Certificate Server Conformance
193 Clause, Basic Authentication Suite}.

- 194 **4.11 Discover Versions TLS 1.2 Authentication Server Profile**
195 A profile that consists of the tuple {Discover Versions Server Conformance Clause, TLS 1.2
196 Authentication Suite}
- 197 **4.12 Baseline Server TLS 1.2 Authentication KMIP Profile**
198 A profile that consists of the tuple {Baseline Server Conformance Clause, TLS 1.2 Authentication Suite}
- 199 **4.13 Secret Data Server TLS 1.2 Authentication KMIP Profile**
200 A profile that consists of the tuple {Secret Data Server Conformance Clause, TLS 1.2 Authentication
201 Suite}
- 202 **4.14 Symmetric Key Store and Server TLS 1.2 Authentication KMIP
203 Profile**
204 A profile that consists of the tuple {Basic Symmetric Key Store and Server Conformance Clause, TLS 1.2
205 Authentication Suite}
- 206 **4.15 Symmetric Key Foundry and Server TLS 1.2 Authentication KMIP
207 Profile**
208 A profile that consists of the tuple {Basic Symmetric Key Foundry and Server Conformance Clause, TLS
- 209 **4.16 Asymmetric Key Store Server TLS 1.2 Authentication KMIP
210 Profile**
211 A profile that consists of the tuple {Asymmetric Key Store Server Conformance Clause, TLS 1.2
212 Authentication Suite}
- 213 **4.17 Asymmetric Key and Certificate Store Server TLS 1.2
214 Authentication KMIP Profile**
215 A profile that consists of the tuple {Asymmetric Key Foundry and Certificate Store Server Conformance
216 Clause, TLS 1.2 Authentication Suite}
- 217 **4.18 Asymmetric Key Foundry and Server TLS 1.2 Authentication
218 KMIP Profile**
219 A profile that consists of the tuple {Asymmetric Key Foundry and Server Conformance Clause, TLS 1.2
220 Authentication Suite}
- 221 **4.19 Certificate Server TLS 1.2 Authentication KMIP Profile**
222 A profile that consists of the tuple {Certificate Server Conformance Clause, TLS 1.2 Authentication Suite}
- 223 **4.20 Asymmetric Key Foundry and Certificate Server TLS 1.2
224 Authentication KMIP Profile**
225 A profile that consists of the tuple {Asymmetric Key Foundry and Certificate Store Server Conformance
226 Clause, TLS 1.2 Authentication Suite}

- 227 **4.21 Basic Discover Versions Client KMIP Profile**
228 A profile that consists of the tuple {Discover Versions Client Conformance Clause, Basic Authentication
229 Suite}
- 230 **4.22 Basic Baseline Client KMIP Profile**
231 A profile that consists of the tuple {Baseline Client Conformance Clause, Basic Authentication Suite}
- 232 **4.23 Basic Secret Data Client KMIP Profile**
233 A profile that consists of the tuple {Secret Data Client Conformance Clause, Basic Authentication Suite}
- 234 **4.24 Basic Symmetric Key Store Client KMIP Profile**
235 A profile that consists of the tuple {Basic Symmetric Key Store Client Conformance Clause, Basic
236 Authentication Suite}
- 237 **4.25 Basic Symmetric Key Foundry Client KMIP Profile**
238 A profile that consists of the tuple {Basic Symmetric Key Foundry Client Conformance Clause, Basic
239 Authentication Suite}
- 240 **4.26 Basic Asymmetric Key Store Client KMIP Profile**
241 A profile that consists of the tuple {Basic Asymmetric Key Store Client Conformance Clause, Basic
242 Authentication Suite}
- 243 **4.27 Basic Asymmetric Key and Certificate Store Client KMIP Profile**
244 A profile that consists of the tuple {Basic Asymmetric Key and Certificate Store Client Conformance
245 Clause, Basic Authentication Suite}
- 246 **4.28 Basic Asymmetric Key Foundry Client KMIP Profile**
247 A profile that consists of the tuple {Basic Asymmetric Key Foundry Client Conformance Clause, Basic
248 Authentication Suite}
- 249 **4.29 Basic Certificate Client KMIP Profile**
250 A profile that consists of the tuple {Basic Certificate Client Conformance Clause, Basic Authentication
251 Suite}
- 252 **4.30 Basic Asymmetric Key Foundry and Certificate Client KMIP
253 Profile**
254 A profile that consists of the tuple {Basic Asymmetric Key Foundry and Certificate Client Conformance
255 Clause, Basic Authentication Suite}
- 256 **4.31 Discover Versions Client TLS 1.2 Authentication KMIP Profile**
257 A profile that consists of the tuple {Discover Versions Client Conformance Clause, Basic Authentication
258 Suite}
- 259 **4.32 Baseline Client TLS 1.2 Authentication KMIP Profile**
260 A profile that consists of the tuple {Baseline Client Conformance Clause, Basic Authentication Suite}

261 **4.33 Secret Data Client TLS 1.2 Authentication KMIP Profile**
262 A profile that consists of the tuple {Secret Data Client Conformance Clause, TLS 1.2 Authentication Suite}

263 **4.34 Symmetric Key Store Client TLS 1.2 Authentication KMIP Profile**
264 A profile that consists of the tuple {Basic Symmetric Key Store Client Conformance Clause, TLS 1.2
265 Authentication Suite}

266 **4.35 Symmetric Key Foundry Client TLS 1.2 Authentication KMIP
267 Profile**
268 A profile that consists of the tuple {Basic Symmetric Key Foundry and Server Conformance Clause, TLS
269 1.2 Authentication Suite}

270 **4.36 Asymmetric Key Store Client TLS 1.2 Authentication KMIP Profile**
271 A profile that consists of the tuple {Basic Asymmetric Key Store Client Conformance Clause, TLS 1.2
272 Authentication Suite}

273 **4.37 Asymmetric Key and Certificate Store Client TLS 1.2
274 Authentication KMIP Profile**
275 A profile that consists of the tuple {Basic Asymmetric Key and Certificate Store Client Conformance
276 Clause, TLS 1.2 Authentication Suite}

277 **4.38 Asymmetric Key Foundry Client TLS 1.2 Authentication KMIP
278 Profile**
279 A profile that consists of the tuple {Basic Asymmetric Key Foundry Client Conformance Clause, TLS 1.2
280 Authentication Suite}

281 **4.39 Certificate Client TLS 1.2 Authentication KMIP Profile**
282 A profile that consists of the tuple {Basic Certificate Client Conformance Clause, TLS 1.2 Authentication
283 Suite}

284 **4.40 Asymmetric Key Foundry and Certificate Client TLS 1.2
285 Authentication KMIP Profile**
286 A profile that consists of the tuple {Basic Asymmetric Key Foundry and Certificate Client Conformance
287 Clause, TLS 1.2 Authentication Suite}

288 **4.41 Storage Client KMIP Profile**
289 A profile that consists of the tuple {Storage Client Conformance Clause, Basic Authentication Suite}

290 **4.42 Storage Client TLS 1.2 Authentication KMIP Profile**
291 A profile that consists of the tuple {Storage Client Conformance Clause, TLS 1.2 Authentication Suite}
292
293
294

295 5 Conformance Clauses

296 The following subsections describe currently-defined profiles related to the use of KMIP.

297 5.1 Discover Versions Server Clause

298 This proposal builds on the KMIP server conformance clauses to provide the most basic functionality that
299 would be expected of a conformant KMIP server – the ability to provide the server version.

300 5.1.1. Implementation Conformance

301 An implementation is a conforming Discover Versions Server Clause if it meets the conditions as outlined
302 in the following section.

303 5.1.2 Conformance of a Discover Versions Server

304 An implementation conforms to this specification as a Discover Versions Server if it meets the following
305 conditions:

- 306 1. Supports the conditions required by the KMIP Server conformance clauses ([KMIP-Spec] 12.1)
- 307 2. Supports the Discover Versions client-to-server operation ([KMIP-Spec] 4.26)
- 308 3. Supports the Query client-to-server operation ([KMIP-Spec] 4.25)

309 5.2 Baseline Server Clause

310 This proposal builds on the KMIP server conformance clauses to provide some of the most basic
311 functionality that would be expected of a conformant KMIP server – the ability to provide information
312 about the server.

313 5.2.1 Implementation Conformance

314 An implementation is a conforming Baseline Server Clause if it meets the conditions as outlined in the
315 following section.

316 5.2.2 Conformance of a KMIP Baseline Server

317 An implementation conforms to this specification as a Baseline Server if it meets the following conditions:

- 318 1. Supports the conditions required by the KMIP Server conformance clauses ([KMIP-Spec] 12.1)
- 319 2. Supports the following objects:
 - 320 a. Attribute ([KMIP-Spec] 2.1.1)
 - 321 b. Credential ([KMIP-Spec] 2.1.2)
 - 322 c. Key Block ([KMIP-Spec] 2.1.3)
 - 323 d. Key Value ([KMIP-Spec] 2.1.4)
 - 324 e. Template-Attribute Structure ([KMIP-Spec] 2.1.8)
- 325 3. Supports the following subsets of attributes:
 - 326 a. Unique Identifier ([KMIP-Spec] 3.1)
 - 327 b. Name ([KMIP-Spec] 3.2)
 - 328 c. Object Type ([KMIP-Spec] 3.3)
 - 329 d. Cryptographic Algorithm ([KMIP-Spec] 3.4)
 - 330 e. Cryptographic Length ([KMIP-Spec] 3.5)
 - 331 f. Cryptographic Parameters ([KMIP-Spec] 3.6)
 - 332 g. Digest ([KMIP-Spec] 3.17)
 - 333 h. Default Operation Policy ([KMIP-Spec] 3.18.2)

- 334 i. Cryptographic Usage Mask ([KMIP-Spec] 3.19)
- 335 j. State ([KMIP-Spec] 3.22)
- 336 k. Initial Date ([KMIP-Spec] 3.23)
- 337 l. Activation Date ([KMIP-Spec] 3.24)
- 338 m. Deactivation Date ([KMIP-Spec] 3.27)
- 339 n. Compromise Occurrence Date ([KMIP-Spec] 3.29)
- 340 o. Compromise Date ([KMIP-Spec] 3.30)
- 341 p. Revocation Reason ([KMIP-Spec] 3.31)
- 342 q. Last Change Date ([KMIP-Spec] 3.38)
- 343
- 344 4. Supports the ID Placeholder ([KMIP-Spec] 4)
- 345 5. Supports the following client-to-server operations:
 - 346 a. Locate ([KMIP-Spec] 4.9)
 - 347 b. Check ([KMIP-Spec] 4.10)
 - 348 c. Get ([KMIP-Spec] 4.11)
 - 349 d. Get Attributes ([KMIP-Spec] 4.12)
 - 350 e. Get Attribute List ([KMIP-Spec] 4.13)
 - 351 f. Add Attribute ([KMIP-Spec] 4.14)
 - 352 g. Modify Attribute ([KMIP-Spec] 4.15)
 - 353 h. Delete Attribute ([KMIP-Spec] 4.16)
 - 354 i. Activate ([KMIP-Spec] 4.19)
 - 355 j. Revoke ([KMIP-Spec] 4.20)
 - 356 k. Destroy ([KMIP-Spec] 4.21)
 - 357 l. Query ([KMIP-Spec] 4.25)
 - 358 m. Discover Versions ([KMIP-Spec] 4.26)
 - 359
- 360 6. Supports the following message contents:
 - 361 a. Protocol Version ([KMIP-Spec] 6.1)
 - 362 b. Operation ([KMIP-Spec] 6.2)
 - 363 c. Maximum Response Size ([KMIP-Spec] 6.3)
 - 364 d. Unique Batch Item ID ([KMIP-Spec] 6.4)
 - 365 e. Time Stamp ([KMIP-Spec] 6.5)
 - 366 f. Asynchronous Indicator ([KMIP-Spec] 6.7)
 - 367 g. Result Status ([KMIP-Spec] 6.9)
 - 368 h. Result Reason ([KMIP-Spec] 6.10)
 - 369 i. Batch Order Option ([KMIP-Spec] 6.12)
 - 370 j. Batch Error Continuation Option ([KMIP-Spec] 6.13)
 - 371 k. Batch Count ([KMIP-Spec] 6.14)
 - 372 l. Batch Item ([KMIP-Spec] 6.15)
 - 373
- 374 7. Supports Message Format ([KMIP-Spec] 7)
- 375 8. Supports Authentication ([KMIP-Spec] 8)
- 376 9. Supports the TTLV encoding ([KMIP-Spec] 9.1)
- 377 10. Supports the transport requirements ([KMIP-Spec] 10)
- 378 11. Supports Error Handling ([KMIP-Spec] 11) for any supported object, attribute, or operation
- 379 12. Optionally supports any clause within [KMIP-Spec] that is not listed above
- 380 13. Optionally supports extensions outside the scope of this standard (e.g., vendor extensions,
- 381 conformance clauses) that do not contradict any KMIP requirements

382

383 5.3 Secret Data Server Clause

384 This proposal builds on the KMIP server conformance clauses to provide some of the most basic
385 functionality that would be expected of a conformant KMIP server – the ability to create, register and get
386 secret data in an interoperable fashion.

387 **5.3.1 Implementation Conformance**

388 An implementation is a conforming Secret Data Server Clause if it meets the conditions as outlined in the
389 following section.

390 **5.3.2 Conformance of a Secret Data Server**

391 An implementation conforms to this specification as a Secret Data Server if it meets the following
392 conditions:

- 393 1. Supports the conditions required by the KMIP Server conformance clauses ([KMIP-Spec] 12.1
394 and Baseline Server conformance clause ([KMIP-Prof] 5.2)
- 395 2. Supports the following additional objects:
 - 396 a. Secret Data ([KMIP-Spec] 2.2.7)
- 397 3. Supports the following client-to-server operations:
 - 398 a. Register ([KMIP-Spec] 4.3)
- 399 4. Supports the following subsets of enumerated attributes:
 - 400 a. Object Type ([KMIP-Spec] 3.3 and 9.1.3.2.12)
 - 401 i. Secret Data
 - 402 b. Secret Data Type ([KMIP-Spec] 2.2.7 and 9.1.3.2.9)
 - 403 i. Password
- 404 5. Supports the following subsets of enumerated objects ([KMIP-Spec] clauses 3 and 9):
 - 405 a. Key Format Type ([KMIP-Spec] 2.1.3 and 9.1.3.2.3)
 - 406 i. Opaque
- 407 6. Optionally supports any clause within [KMIP-Spec] that is not listed above
- 408 7. Optionally supports extensions outside the scope of this standard (e.g., vendor extensions,
409 conformance clauses) that do not contradict any KMIP requirements

410 **5.4 Symmetric Key Store and Server Conformance Clause**

411 This proposal builds on the KMIP server conformance clauses to provide support for symmetric key store
412 and foundry use cases.

413 **5.4.1 Implementation Conformance**

414 An implementation is a conforming KMIP Symmetric Key Store and Server if the implementation meets
415 the conditions as outlined in the following section.

416 **5.4.2 Conformance as a Symmetric Key Store and Server**

417 An implementation conforms to this specification as a Symmetric Key Store and Server if it meets the
418 following conditions:

- 419 1. Supports the conditions required by the KMIP Server conformance clauses. ([KMIP-Spec] 12.1)
420 and Baseline Server conformance clause ([KMIP-Prof] 5.2)
- 421 2. Supports the following additional objects:
 - 422 a. Symmetric Key ([KMIP-Spec] 2.2.2)
- 423 3. Supports the following client-to-server operations:
 - 424 a. Register ([KMIP-Spec] 4.3)
- 425 4. Supports the following attributes:
 - 426 a. Process Start Date ([KMIP-Spec] 3.25)
 - 427 b. Protect Stop Date ([KMIP-Spec] 3.26)

- 428 5. Supports the following subsets of enumerated attributes:
429 a. Cryptographic Algorithm ([KMIP-Spec] 3.4 and 9.1.3.2.13)
430 i. 3DES
431 ii. AES
432 b. Object Type ([KMIP-Spec] 3.3 and 9.1.3.2.12)
433 i. Symmetric Key
434 6. Supports the following subsets of enumerated objects:
435 a. Key Format Type ([KMIP-Spec] 2.1.3 and 9.1.3.2.3)
436 i. Raw
437 ii. Transparent Symmetric Key
438 7. Optionally supports any clause within [KMIP-Spec] that is not listed above
439 8. Optionally supports extensions outside the scope of this standard (e.g., vendor extensions,
440 conformance clauses) that do not contradict any KMIP requirements

441 5.5 Symmetric Key Foundry and Server Conformance Clause

442 This proposal intends to meet this OASIS requirement by building on the KMIP Server Conformance
443 Clause to provide basic symmetric key services. The intent is to simply allow key creation and serving
444 with very limited key types.

445 5.5.1 Implementation Conformance

446 An implementation is a conforming KMIP Symmetric Key Store and Server if the implementation meets
447 the conditions as outlined in the following section.

448 5.5.2 Conformance as a KMIP Symmetric Key Foundry and Server

449 An implementation conforms to this specification as a KMIP Symmetric Key Foundry and Server if it
450 meets the following conditions:

- 451 1. Supports the conditions required by the KMIP Server conformance clauses. ([KMIP-Spec] 12.1)
452 and Baseline Server conformance clause ([KMIP-Prof] 5.2)
453 2. Supports the following additional objects
454 a. Symmetric Key ([KMIP-Spec] 2.2.2)
455 3. Supports the following client-to-server operations:
456 a. Create ([KMIP-Spec] 4.1)
457 4. Supports the following attributes:
458 a. Process Start Date ([KMIP-Spec] 3.25)
459 b. Protect Stop Date ([KMIP-Spec] 3.26)
460 5. Supports the following subsets of enumerated attributes:
461 a. Cryptographic Algorithm ([KMIP-Spec] 3.4 and 9.1.3.2.13)
462 i. 3DES
463 ii. AES
464 b. Object Type ([KMIP-Spec] 3.3 and 9.1.3.2.12)
465 i. Symmetric Key
466 6. Supports the following subsets of enumerated objects:
467 a. Key Format Type ([KMIP-Spec] 2.1.3 and 9.1.3.2.3)
468 i. Raw

- 469 ii. Transparent Symmetric Key
470 7. Optionally supports any clause within [KMIP-Spec] that is not listed above
471 8. Optionally supports extensions outside the scope of this standard (e.g., vendor extensions,
472 conformance clauses) that do not contradict any KMIP requirements
473

474 **5.6 Asymmetric Key Store Server Conformance Clauses**

475 This proposal intends to meet this OASIS requirement by building on the KMIP Server Conformance
476 Clauses to allow asymmetric key pairs generated external to the key server to be vaulted by a key server.
477 The intent is to simply support key registration for a very limited number of key types.

478 **5.6.1 Implementation Conformance**

479 An implementation is a conforming KMIP Asymmetric Key Store Server if the implementation meets the
480 conditions as outlined in the following section.

481 **5.6.2 Conformance as an Asymmetric Key Store Server**

482 An implementation conforms to this specification as a KMIP Asymmetric Key Store Server if it meets the
483 following conditions:

- 484 1. Supports the conditions required by the KMIP Server conformance clauses ([KMIP-Spec] 12.1
485 and Baseline Server conformance clause ([KMIP-Prof] 5.2))
- 486 2. Supports the following additional objects:
- 487 a. Public Key ([KMIP-Spec] 2.2.3)
- 488 b. Private Key ([KMIP-Spec] 2.2.4)
- 489 3. Supports the following client-to-server operations:
- 490 a. Register ([KMIP-Spec] 4.3)
- 491 4. Supports the following subset of enumerated attributes:
- 492 a. Object Type ([KMIP-Spec] 3.3 and 9.1.3.2.12)
- 493 i. Public Key
- 494 ii. Private Key
- 495 b. Cryptographic Algorithm ([KMIP-Spec] 3.4 and 9.1.3.2.13)
- 496 i. RSA
- 497 c. Link ([KMIP-Spec] 3.35 and 9.1.3.2.20)
- 498 i. Public Key Link
- 499 ii. Private Key Link
- 500 5. Supports the following subset of enumerated objects:
- 501 a. Key Format Type ([KMIP-Spec] 2.1.3 and 9.1.3.2.3)
- 502 i. PKCS#1
- 503 6. Optionally supports any clause within [KMIP-Spec] that is not listed above
- 504 7. Optionally supports extensions outside the scope of this standard (e.g., vendor extensions,
505 Conformance Clauses) that do not contradict any requirements within this standard
506

507 **5.7 Asymmetric Key and Certificate Store Server Conformance**
508 **Clauses**

509 This proposal intends to meet this OASIS requirement by building on the KMIP Server Conformance
510 Clauses to allow asymmetric key pairs and certificates generated external to the key server to be vaulted
511 by a key server. The intent is to simply support key and certificate registration for a very limited number
512 of key types.

513 **5.7.1 Implementation Conformance**

514 An implementation is a conforming KMIP Asymmetric Key and Certificate Store Server if the
515 implementation meets the conditions as outlined in the following section.

516 **5.7.2 Conformance as a Asymmetric Key and Certificate Store Server**

517 An implementation conforms to this specification as a KMIP Asymmetric Key and Certificate Store Server
518 if it meets the following conditions:

- 519 1. Supports the conditions required by the KMIP Server conformance clauses ([KMIP-Spec] 12.1)
520 and Baseline Server conformance clause ([KMIP-Prof] 5.2)
- 521 2. Supports the following subsets of additional objects:
 - 522 a. Certificate ([KMIP-Spec] 2.2.1)
 - 523 b. Public Key ([KMIP-Spec] 2.2.3)
 - 524 c. Private Key ([KMIP-Spec] 2.2.4)
- 525 3. Supports the following client-to-server operations:
 - 526 a. Register ([KMIP-Spec] 4.3)
- 527 4. Supports the following subset of enumerated attributes:
 - 528 a. Object Type ([KMIP-Spec] 3.3 and 9.1.3.2.12)
 - 529 i. Certificate
 - 530 ii. Public Key
 - 531 iii. Private Key
 - 532 b. Cryptographic Algorithm ([KMIP-Spec] 3.4 and 9.1.3.2.13)
 - 533 i. RSA
 - 534 c. Certificate Type ([KMIP-Spec] 3.8 and 9.1.3.2.6)
 - 535 i. X.509
 - 536 d. X.509 Certificate Identifier ([KMIP-Spec] 3.10)
 - 537 e. X.509 Certificate Subject ([KMIP-Spec] 3.11)
 - 538 f. X.509 Certificate Issuer ([KMIP-Spec] 3.12)
 - 539 g. Link ([KMIP-Spec] 3.35 and 9.1.3.2.20)
 - 540 a. Certificate Link
 - 541 b. Public Key Link
 - 542 c. Private Key Link
- 543 5. Supports the following subset of enumerated objects:
 - 544 d. Key Format Type ([KMIP-Spec] 2.1.3 and 9.1.3.2.3)
 - 545 i. PKCS#1
 - 546 ii. X.509
- 547 6. Optionally supports any clause within [KMIP-Spec] that is not listed above

548 7. Optionally supports extensions outside the scope of this standard (e.g., vendor extensions,
549 Conformance Clauses) that do not contradict any requirements within this standard

550 **5.8 Asymmetric Key Foundry and Server Conformance Clauses**

551 This proposal intends to meet this OASIS requirement by building on the KMIP Server Conformance
552 Clauses to provide basic asymmetric key services for central key generation (by the key server). The
553 intent is to simply allow key creation and serving with very limited key types.

554 **5.8.1 Implementation Conformance**

555 An implementation is a conforming KMIP Asymmetric Key Foundry and Server if the implementation
556 meets the conditions as outlined in the following section.

557 **5.8.2 Conformance as a Asymmetric Key Foundry and Server**

558 An implementation conforms to this specification as a KMIP Asymmetric Key Foundry and Server if it
559 meets the following conditions:

- 560 1. Supports the conditions required by the KMIP Server conformance clauses ([KMIP-Spec] 12.1)
561 and Baseline Server conformance clause ([KMIP-Prof] 5.2)
- 562 2. Supports the following additional objects:
 - 563 a. Public Key ([KMIP-Spec] 2.2.3)
 - 564 b. Private Key ([KMIP-Spec] 2.2.4)
- 565 3. Supports the following client-to-server operations:
 - 566 a. Create Key Pair ([KMIP-Spec] 4.2)
 - 567 b. Re-key Key Pair ([KMIP-Spec] 4.5)
- 568 4. Supports the following subset of enumerated attributes:
 - 569 a. Object Type ([KMIP-Spec] 3.3 and 9.1.3.2.12)
 - 570 i. Public Key
 - 571 ii. Private Key
 - 572 b. Cryptographic Algorithm ([KMIP-Spec] 3.4 and 9.1.3.2.13)
 - 573 i. RSA
 - 574 c. Link ([KMIP-Spec] 3.35 and 9.1.3.2.20)
 - 575 i. Public Key Link
 - 576 ii. Private Key Link
 - 577 iii. Replacement Object Link
 - 578 iv. Replaced Object Link
- 579 5. Supports the following subset of enumerated objects:
 - 580 c. Key Format Type ([KMIP-Spec] 2.1.3 and 9.1.3.2.3)
 - 581 i. PKCS#1
 - 582 ii. Transparent RSA private key ([KMIP-Spec] 2.1.7.4)
 - 583 iii. Transparent RSA public key ([KMIP-Spec] 2.1.7.5)
- 584 6. Optionally supports any clause within [KMIP-Spec] that is not listed above
- 585 7. Optionally supports extensions outside the scope of this standard (e.g., vendor extensions,
586 Conformance Clauses) that do not contradict any requirements within this standard

587 **5.9 Certificate Server Conformance Clauses**

588 This proposal intends to meet this OASIS requirement by building on the KMIP Server Conformance
589 Clauses to provide basic asymmetric key services for local key generation (external to the key server) and
590 certification via a key server.

591 **5.9.1 Implementation Conformance**

592 An implementation is a conforming KMIP Certificate Server if the implementation meets the conditions as
593 outlined in the following section.

594 **5.9.2 Conformance as a Certificate Server**

595 An implementation conforms to this specification as a KMIP Certificate Server if it meets the following
596 conditions:

- 597 1. Supports the conditions required by the KMIP Server conformance clauses ([KMIP-Spec] 12.1)
598 and Baseline Server conformance clause ([KMIP-Prof] 5.2)
- 599 2. Supports the following additional objects:
 - 600 a. Certificate ([KMIP-Spec] 2.2.1)
 - 601 b. Public Key ([KMIP-Spec] 2.2.3)
 - 602 c. Private Key ([KMIP-Spec] 2.2.4)
- 603 3. Supports the following client-to-server operations:
 - 604 a. Certify ([KMIP-Spec] 4.7)
 - 605 b. Re-Certify ([KMIP-Spec] 4.8)
- 606 4. Supports the following subset of enumerated attributes:
 - 607 a. Object Type ([KMIP-Spec] 3.3 and 9.1.3.2.12)
 - 608 i. Certificate
 - 609 ii. Public Key
 - 610 iii. Private Key
 - 611 b. Cryptographic Algorithm ([KMIP-Spec] 3.4 and 9.1.3.2.13)
 - 612 i. RSA
 - 613 c. Certificate Type ([KMIP-Spec] 3.8 and 9.1.3.2.6)
 - 614 i. X.509
 - 615 d. X.509 Certificate Identifier ([KMIP-Spec] 3.10)
 - 616 e. X.509 Certificate Subject ([KMIP-Spec] 3.11)
 - 617 f. X.509 Certificate Issuer ([KMIP-Spec] 3.12)
 - 618 g. Link ([KMIP-Spec] 3.35 and 9.1.3.2.20)
 - 619 i. Certificate Link
 - 620 ii. Public Key Link
 - 621 iii. Private Key Link
 - 622 iv. Replacement Object Link
 - 623 v. Replaced Object Link
 - 624 h. Certificate Request Type ([KMIP-Spec] 4.7, 4.8 and 9.1.3.2.22)
 - 625 i. PKCS#10
 - 626 ii. PEM
- 627 5. Supports the following subsets of enumerated objects:
 - 628 a. Key Format Type ([KMIP-Spec] 2.1.3 and 9.1.3.2.3)

- 629 i. PKCS#1
- 630 ii. X.509
- 631 6. Optionally supports any clause within [KMIP-Spec] that is not listed above
- 632 7. Optionally supports extensions outside the scope of this standard (e.g., vendor extensions,
- 633 Conformance Clauses) that do not contradict any requirements within this standard

634 **5.10 Asymmetric Key Foundry and Certificate Server Conformance** 635 **Clauses**

636 This proposal intends to meet this OASIS requirement by building on the KMIP Server Conformance
637 Clauses to provide basic asymmetric key services for central key generation (by the key server). The
638 intent is to simply allow key and certificate creation and serving with very limited key types.

639 **5.10.1 Implementation Conformance**

640 An implementation is a conforming KMIP Asymmetric Key Foundry and Server if the implementation
641 meets the conditions as outlined in the following section.

642 **5.10.2 Conformance as a Asymmetric Key Foundry and Certificate Server**

643 An implementation conforms to this specification as a KMIP Asymmetric Key Foundry and Certificate
644 Server (Central Generation) if it meets the following conditions:

- 645 1. Supports the conditions required by the KMIP Server conformance clauses ([KMIP-Spec] 12.1)
- 646 and Baseline Server conformance clause ([KMIP-Prof] 5.2)
- 647 2. Supports the following additional objects:
 - 648 a. Certificate ([KMIP-Spec] 2.2.1)
 - 649 b. Public Key ([KMIP-Spec] 2.2.3)
 - 650 c. Private Key ([KMIP-Spec] 2.2.4)
- 651 3. Supports the following client-to-server operations:
 - 652 a. Create Key Pair ([KMIP-Spec] 4.2)
 - 653 b. Re-key Key Pair ([KMIP-Spec] 4.5)
 - 654 c. Certify ([KMIP-Spec] 4.7)
 - 655 d. Re-Certify ([KMIP-Spec] 4.8)
- 656 4. Supports the following subset of enumerated attributes:
 - 657 a. Object Type ([KMIP-Spec] 3.3 and 9.1.3.2.12)
 - 658 i. Certificate
 - 659 ii. Public Key
 - 660 iii. Private Key
 - 661 b. Cryptographic Algorithm ([KMIP-Spec] 3.4 and 9.1.3.2.13)
 - 662 i. RSA
 - 663 c. Certificate Type ([KMIP-Spec] 3.8 and 9.1.3.2.6)
 - 664 i. X.509
 - 665 d. X.509 Certificate Identifier ([KMIP-Spec] 3.10)
 - 666 e. X.509 Certificate Subject ([KMIP-Spec] 3.11)
 - 667 f. X.509 Certificate Issuer ([KMIP-Spec] 3.12)
 - 668 g. Link ([KMIP-Spec] 3.35 and 9.1.3.2.20)
 - 669 i. Certificate Link

- 670 ii. Public Key Link
- 671 iii. Private Key Link
- 672 iv. Replacement Object Link
- 673 v. Replaced Object Link
- 674 h. Certificate Request Type ([KMIP-Spec] 4.7, 4.8 and 9.1.3.2.22)
- 675 i. PKCS#10
- 676 ii. PEM
- 677 5. Supports the following subset of enumerated objects:
- 678 d. Key Format Type ([KMIP-Spec] 2.1.3 and 9.1.3.2.3)
- 679 i. PKCS#1
- 680 ii. X.509
- 681 iii. Transparent RSA private key ([KMIP-Spec] 2.1.7.4)
- 682 iv. Transparent RSA public key ([KMIP-Spec] 2.1.7.4)
- 683 6. Optionally supports any clause within [KMIP-Spec] that is not listed above
- 684 7. Optionally supports extensions outside the scope of this standard (e.g., vendor extensions,
- 685 Conformance Clauses) that do not contradict any requirements within this standard

686 **5.11 Discover Versions Client Clause**

687 This proposal builds on the KMIP client conformance clauses to provide the most basic functionality that
688 would be expected of a conformant KMIP client – the ability to request the server version.

689 **5.11.1 Implementation Conformance**

690 An implementation is a conforming Discover Versions Client Clause if it meets the conditions as outlined
691 in the following section.

692 **5.11.2 Conformance of a Discover Versions Client**

693 An implementation conforms to this specification as a Discover Versions Server if it meets the following
694 conditions:

- 695 1. Supports the conditions required by the KMIP Client conformance clauses ([KMIP-Spec] 12.2)
- 696 2. Supports the Discover Versions client-to-server operation ([KMIP-Spec] 4.26)
- 697 3. Supports the Query client-to-server operation ([KMIP-Spec] 4.25)

698 **5.12 Baseline Client Clause**

699 This proposal builds on the KMIP client conformance clauses to provide some of the most basic
700 functionality that would be expected of a conformant KMIP client – the ability to request information about
701 the server.

702 **5.12.1 Implementation Conformance**

703 An implementation is a conforming Baseline Client Clause if it meets the conditions as outlined in the
704 following section.

705 **5.12.2 Conformance of a KMIP Baseline Client**

706 An implementation conforms to this specification as a Baseline Client if it meets the following conditions:

- 707 1. Supports the conditions required by the KMIP Client conformance clauses ([KMIP-Spec] 12.2)
- 708 2. Supports the following objects:

- 709 a. Attribute ([KMIP-Spec] 2.1.1)
- 710 b. Credential ([KMIP-Spec] 2.1.2)
- 711 c. Key Block ([KMIP-Spec] 2.1.3)
- 712 d. Key Value ([KMIP-Spec] 2.1.4)
- 713 e. Template-Attribute Structure ([KMIP-Spec] 2.1.8)
- 714 3. Supports the following subsets of attributes:
 - 715 a. Unique Identifier ([KMIP-Spec] 3.1)
 - 716 b. Name ([KMIP-Spec] 3.2)
 - 717 c. Object Type ([KMIP-Spec] 3.3)
 - 718 d. Cryptographic Algorithm ([KMIP-Spec] 3.4)
 - 719 e. Cryptographic Length ([KMIP-Spec] 3.5)
 - 720 f. Cryptographic Parameters ([KMIP-Spec] 3.6)
 - 721 g. Digest ([KMIP-Spec] 3.17)
 - 722 h. Default Operation Policy ([KMIP-Spec] 3.18.2)
 - 723 i. Cryptographic Usage Mask ([KMIP-Spec] 3.19)
 - 724 j. State ([KMIP-Spec] 3.22)
 - 725 k. Initial Date ([KMIP-Spec] 3.23)
 - 726 l. Activation Date ([KMIP-Spec] 3.24)
 - 727 m. Deactivation Date ([KMIP-Spec] 3.27)
 - 728 n. Compromise Occurrence Date ([KMIP-Spec] 3.29)
 - 729 o. Compromise Date ([KMIP-Spec] 3.30)
 - 730 p. Revocation Reason ([KMIP-Spec] 3.31)
 - 731 q. Last Change Date ([KMIP-Spec] 3.38)
- 732 4. Supports the ID Placeholder ([KMIP-Spec] 4)
- 733 5. Supports the following client-to-server operations:
 - 734 a. Locate ([KMIP-Spec] 4.9)
 - 735 b. Check ([KMIP-Spec] 4.10)
 - 736 c. Get ([KMIP-Spec] 4.11)
 - 737 d. Get Attributes ([KMIP-Spec] 4.12)
 - 738 e. Get Attribute List ([KMIP-Spec] 4.13)
 - 739 f. Add Attribute ([KMIP-Spec] 4.14)
 - 740 g. Modify Attribute ([KMIP-Spec] 4.15)
 - 741 h. Delete Attribute ([KMIP-Spec] 4.16)
 - 742 i. Activate ([KMIP-Spec] 4.19)
 - 743 j. Revoke ([KMIP-Spec] 4.20)
 - 744 k. Destroy ([KMIP-Spec] 4.21)
 - 745 l. Query ([KMIP-Spec] 4.25)
 - 746 m. Discover Versions ([KMIP-Spec] 4.26)
- 747 6. Supports the following message contents:
 - 748 a. Protocol Version ([KMIP-Spec] 6.1)
 - 749 b. Operation ([KMIP-Spec] 6.2)
 - 750 c. Maximum Response Size ([KMIP-Spec] 6.3)
 - 751 d. Unique Batch Item ID ([KMIP-Spec] 6.4)
 - 752 e. Time Stamp ([KMIP-Spec] 6.5)
 - 753 f. Asynchronous Indicator ([KMIP-Spec] 6.7)
 - 754 g. Result Status ([KMIP-Spec] 6.9)
 - 755 h. Result Reason ([KMIP-Spec] 6.10)
 - 756 i. Batch Order Option ([KMIP-Spec] 6.12)
 - 757 j. Batch Error Continuation Option ([KMIP-Spec] 6.13)
 - 758 k. Batch Count ([KMIP-Spec] 6.14)
 - 759 l. Batch Item ([KMIP-Spec] 6.15)
- 760 7. Supports Message Format ([KMIP-Spec] 7)
- 761 8. Supports Authentication ([KMIP-Spec] 8)
- 762
- 763
- 764

- 765 9. Supports the TTLV encoding ([KMIP-Spec] 9.1)
766 10. Supports the transport requirements ([KMIP-Spec] 10)
767 11. Supports Error Handling ([KMIP-Spec] 11) for any supported object, attribute, or operation
768 12. Optionally supports any clause within [KMIP-Spec] that is not listed above.
769 13. Optionally supports extensions outside the scope of this standard (e.g., vendor extensions,
770 conformance clauses) that do not contradict any KMIP requirements

771 **5.13 Secret Data Client Clause**

772 This proposal builds on the KMIP client conformance clauses to provide some of the most basic
773 functionality that would be expected of a conformant KMIP client – the ability to create, register and get
774 secret data in an interoperable fashion.

775 **5.13.1 Implementation Conformance**

776 An implementation is a conforming Secret Data Client Clause if it meets the conditions as outlined in the
777 following section.

778 **5.13.2 Conformance of a Secret Data Client**

779 An implementation conforms to this specification as a Secret Data Client if it meets the following
780 conditions:

- 781 1. Supports the conditions required by the KMIP Client conformance clauses ([KMIP-Spec] 12.2)
782 and Baseline Client conformance clause ([KMIP-Prof] 5.12)
783 2. Supports the KMIP Baseline Client conformance clauses.
784 3. Supports the following additional objects:
785 a. Secret Data ([KMIP-Spec] 2.2.7)
786 4. Supports the following client-to-server operations:
787 a. Register ([KMIP-Spec] 4.3)
788 5. Supports the following subsets of enumerated attributes:
789 a. Object Type ([KMIP-Spec] 3.3 and 9.1.3.2.12)
790 i. Secret Data
791 b. Secret Data Type ([KMIP-Spec] 2.2.7 and 9.1.3.2.9)
792 i. Password
793 6. Supports the following subsets of enumerated objects ([KMIP-Spec] clauses 3 and 9):
794 a. Key Format Type ([KMIP-Spec] 2.1.3 and 9.1.3.2.3)
795 i. Opaque
796 7. Optionally supports any clause within [KMIP-Spec] specification that is not listed above
797 8. Optionally supports extensions outside the scope of this standard (e.g., vendor extensions,
798 conformance clauses) that do not contradict any KMIP requirements

799 **5.14 Symmetric Key Store Client Conformance Clause**

800 This proposal builds on the KMIP client conformance clauses to provide support for symmetric key store
801 and foundry use cases.

802 **5.14.1 Implementation Conformance**

803 An implementation is a conforming KMIP Symmetric Key Store Client if the implementation meets the
804 conditions as outlined in the following section.

805 **5.14.2 Conformance as a Symmetric Key Store Client**

806 An implementation conforms to this specification as a Basic Symmetric Key Store Client if it meets the
807 following conditions:

- 808 1. Supports the conditions required by the KMIP Client conformance clauses. ([KMIP-Spec] 12.2)
809 and Baseline Client conformance clause ([KMIP-Prof] 5.12)
- 810 2. Supports the following additional objects:
 - 811 a. Symmetric Key ([KMIP-Spec] 2.2.2)
- 812 3. Supports the following client-to-server operations:
 - 813 a. Register ([KMIP-Spec] 4.3)
- 814 4. Supports the following attributes:
 - 815 a. Process Start Date ([KMIP-Spec] 3.25)
 - 816 b. Protect Stop Date ([KMIP-Spec] 3.26)
- 817 5. Supports the following subsets of enumerated attributes:
 - 818 a. Cryptographic Algorithm ([KMIP-Spec] 3.4 and 9.1.3.2.13)
 - 819 i. 3DES
 - 820 ii. AES
 - 821 b. Object Type ([KMIP-Spec] 3.3 and 9.1.3.2.12)
 - 822 i. Symmetric Key
- 823 6. Supports the following subsets of enumerated objects:
 - 824 a. Key Format Type ([KMIP-Spec] 2.1.3 and 9.1.3.2.3)
 - 825 i. Raw
 - 826 ii. Transparent Symmetric Key
- 827 7. Optionally supports any clause within [KMIP-Spec] that is not listed above
- 828 8. Optionally supports extensions outside the scope of this standard (e.g., vendor extensions,
829 conformance clauses) that do not contradict any KMIP requirements

830 **5.15 Symmetric Key Foundry Client Conformance Clause**

831 This proposal intends to meet this OASIS requirement by building on the KMIP Client Conformance
832 Clause to provide basic symmetric key services. The intent is to simply allow key creation and serving
833 with very limited key types.

834 **5.15.1 Implementation Conformance**

835 An implementation is a conforming KMIP Symmetric Key Foundry Client if the implementation meets the
836 conditions as outlined in the following section.

837 **5.15.2 Conformance as a KMIP Symmetric Key Foundry Client**

838 An implementation conforms to this specification as a KMIP Symmetric Key Foundry Client if it meets the
839 following conditions:

- 840 1. Supports the conditions required by the KMIP Client conformance clauses. ([KMIP-Spec] 12.2)
841 and Baseline Client conformance clause ([KMIP-Prof] 5.12)
- 842 2. Supports the following additional objects
 - 843 a. Symmetric Key ([KMIP-Spec] 2.2.2)
- 844 3. Supports the following client-to-server operations:
 - 845 a. Create ([KMIP-Spec] 4.1)

- 846 4. Supports the following attributes:
- 847 a. Process Start Date ([KMIP-Spec] 3.25)
- 848 b. Protect Stop Date ([KMIP-Spec] 3.26)
- 849 5. Supports the following subsets of enumerated attributes:
- 850 a. Cryptographic Algorithm ([KMIP-Spec] 3.4 and 9.1.3.2.13)
- 851 i. 3DES
- 852 ii. AES
- 853 b. Object Type ([KMIP-Spec] 3.3 and 9.1.3.2.12)
- 854 i. Symmetric Key
- 855 6. Supports the following subsets of enumerated objects:
- 856 a. Key Format Type ([KMIP-Spec] 2,1,3 and 9.1.3.2.3)
- 857 i. Raw
- 858 ii. Transparent Symmetric Key
- 859 7. Optionally supports any clause within [KMIP-Spec] that is not listed above
- 860 8. Optionally supports extensions outside the scope of this standard (e.g., vendor extensions,
- 861 conformance clauses) that do not contradict any KMIP requirements
- 862

863 5.16 Asymmetric Key Store Client Conformance Clauses

864 This proposal intends to meet this OASIS requirement by building on the KMIP Client Conformance
865 Clauses to allow asymmetric key pairs generated external to the key server to be vaulted by a key server.
866 The intent is to simply support key registration for a very limited number of key types.

867 5.16.1 Implementation Conformance

868 An implementation is a conforming KMIP Asymmetric Key Store Client if the implementation meets the
869 conditions as outlined in the following section.

870 5.16.2 Conformance as a Asymmetric Key Store Client

871 An implementation conforms to this specification as a KMIP Asymmetric Key Store Client if it meets the
872 following conditions:

- 873 1. Supports the conditions required by the KMIP Client conformance clauses ([KMIP-Spec] 12.2)
- 874 and Baseline Client conformance clause ([KMIP-Prof] 5.12)
- 875 2. Supports the following additional objects:
- 876 a. Public Key ([KMIP-Spec] 2.2.3)
- 877 b. Private Key ([KMIP-Spec] 2.2.4)
- 878 3. Supports the following client-to-server operations:
- 879 a. Register ([KMIP-Spec] 4.3)
- 880 4. Supports the following subset of enumerated attributes:
- 881 a. Object Type ([KMIP-Spec] 3.3 and 9.1.3.2.12)
- 882 i. Public Key
- 883 ii. Private Key
- 884 b. Cryptographic Algorithm ([KMIP-Spec] 3.4 and 9.1.3.2.13)
- 885 i. RSA
- 886 c. Link ([KMIP-Spec] 3.35 and 9.1.3.2.20)

- 887 i. Public Key Link
- 888 ii. Private Key Link
- 889 5. Supports the following subset of enumerated objects:
 - 890 a. Key Format Type ([KMIP-Spec] 2.1.3 and 9.1.3.2.3)
 - 891 i. Raw
 - 892 ii. PKCS#1
- 893 6. Optionally supports any clause within [KMIP-Spec] that is not listed above
- 894 7. Optionally supports extensions outside the scope of this standard (e.g., vendor extensions,
- 895 Conformance Clauses) that do not contradict any requirements within this standard
- 896

897 **5.17 Asymmetric Key and Certificate Store Client Conformance**

898 **Clauses**

899 This proposal intends to meet this OASIS requirement by building on the KMIP Client Conformance
900 Clauses to allow asymmetric key pairs and certificates generated external to the key server to be vaulted
901 by a key server. The intent is to simply support key and certificate registration for a very limited number
902 of key types.

903 **5.17.1 Implementation Conformance**

904 An implementation is a conforming KMIP Asymmetric Key and Certificate Store Client if the
905 implementation meets the conditions as outlined in the following section.

906 **5.17.2 Conformance as an Asymmetric Key and Certificate Store Client**

907 An implementation conforms to this specification as a KMIP Asymmetric Key and Certificate Store Client if
908 it meets the following conditions:

- 909 1. Supports the conditions required by the KMIP Client conformance clauses ([KMIP-Spec] 12.2)
- 910 and Baseline Client conformance clause ([KMIP-Prof] 5.12)
- 911 2. Supports the following subsets of additional objects:
 - 912 a. Certificate ([KMIP-Spec] 2.2.1)
 - 913 b. Public Key ([KMIP-Spec] 2.2.3)
 - 914 c. Private Key ([KMIP-Spec] 2.2.4)
- 915 3. Supports the following client-to-server operations:
 - 916 a. Register ([KMIP-Spec] 4.3)
- 917 4. Supports the following subset of enumerated attributes:
 - 918 a. Object Type ([KMIP-Spec] 3.3 and 9.1.3.2.12)
 - 919 i. Certificate
 - 920 ii. Public Key
 - 921 iii. Private Key
 - 922 b. Cryptographic Algorithm ([KMIP-Spec] 3.4 and 9.1.3.2.13)
 - 923 i. RSA
 - 924 c. Certificate Type ([KMIP-Spec] 3.8 and 9.1.3.2.6)
 - 925 i. X.509
 - 926 d. X.509 Certificate Identifier ([KMIP-Spec] 3.10)
 - 927 e. X.509 Certificate Subject ([KMIP-Spec] 3.11)
 - 928 f. X.509 Certificate Issuer ([KMIP-Spec] 3.12)

- 929 g. Link ([KMIP-Spec] 3.35 and 9.1.3.2.20)
- 930 a. Certificate Link
- 931 b. Public Key Link
- 932 c. Private Key Link
- 933 5. Supports the following subset of enumerated objects:
- 934 a. Key Format Type ([KMIP-Spec] 2.1.3 and 9.1.3.2.3)
- 935 i. PKCS#1
- 936 ii. X.509
- 937 6. Optionally supports any clause within [KMIP-Spec] that is not listed above
- 938 7. Optionally supports extensions outside the scope of this standard (e.g., vendor extensions,
- 939 Conformance Clauses) that do not contradict any requirements within this standard

940 **5.18 Asymmetric Key Foundry Client Conformance Clauses**

941 This proposal intends to meet this OASIS requirement by building on the KMIP Client Conformance
942 Clauses to provide basic asymmetric key services for central key generation (by the key server). The
943 intent is to simply allow key creation and serving with very limited key types.

944 **5.18.1 Implementation Conformance**

945 An implementation is a conforming KMIP Asymmetric Key Foundry Client if the implementation meets the
946 conditions as outlined in the following section.

947 **5.18.2 Conformance as an Asymmetric Key Foundry Client**

948 An implementation conforms to this specification as a KMIP Asymmetric Key Foundry Client if it meets the
949 following conditions:

- 950 1. Supports the conditions required by the KMIP Server conformance clauses ([KMIP-Spec] 12.2)
- 951 and Baseline Client conformance clause ([KMIP-Prof] 5.12)
- 952 2. Supports the following additional objects:
- 953 a. Public Key ([KMIP-Spec] 2.2.3)
- 954 b. Private Key ([KMIP-Spec] 2.2.4)
- 955 3. Supports the following client-to-server operations:
- 956 a. Create Key Pair ([KMIP-Spec] 4.2)
- 957 b. Re-key Key Pair ([KMIP-Spec] 4.5)
- 958 4. Supports the following subset of enumerated attributes:
- 959 a. Object Type ([KMIP-Spec] 3.3 and 9.1.3.2.12)
- 960 i. Public Key
- 961 ii. Private Key
- 962 b. Cryptographic Algorithm ([KMIP-Spec] 3.4 and 9.1.3.2.13)
- 963 i. RSA
- 964 c. Link ([KMIP-Spec] 3.35 and 9.1.3.2.20)
- 965 i. Public Key Link
- 966 ii. Private Key Link
- 967 iii. Replacement Object Link
- 968 iv. Replaced Object Link
- 969 5. Supports the following subset of enumerated objects:

- 970 a. Key Format Type ([KMIP-Spec] 2.1.3 and 9.1.3.2.3)
- 971 i. PKCS#1
- 972 ii. Transparent RSA private key ([KMIP-Spec] 2.1.7.4)
- 973 iii. Transparent RSA public key ([KMIP-Spec] 2.1.7.5)
- 974 6. Optionally supports any clause within [KMIP-Spec] that is not listed above
- 975 7. Optionally supports extensions outside the scope of this standard (e.g., vendor extensions,
- 976 Conformance Clauses) that do not contradict any requirements within this standard

977 **5.19 Certificate Client Conformance Clauses**

978 This proposal intends to meet this OASIS requirement by building on the KMIP Client Conformance
979 Clauses to provide basic asymmetric key services for local key generation (external to the key server) and
980 certification via a key server.

981 **5.19.1 Implementation Conformance**

982 An implementation is a conforming KMIP Certificate Client if the implementation meets the conditions as
983 outlined in the following section.

984 **5.19.2 Conformance as a Basic Certificate Client**

985 An implementation conforms to this specification as a KMIP Certificate Client if it meets the following
986 conditions:

- 987 1. Supports the conditions required by the KMIP Client conformance clauses ([KMIP-Spec] 12.2)
- 988 and Baseline Client conformance clause ([KMIP-Prof] 5.12)
- 989 2. Supports the following additional objects:
 - 990 a. Certificate ([KMIP-Spec] 2.2.1)
 - 991 b. Public Key ([KMIP-Spec] 2.2.3)
 - 992 c. Private Key ([KMIP-Spec] 2.2.4)
- 993 3. Supports the following client-to-server operations:
 - 994 a. Certify ([KMIP-Spec] 4.7)
 - 995 b. Re-Certify ([KMIP-Spec] 4.8)
- 996 4. Supports the following subset of enumerated attributes:
 - 997 a. Object Type ([KMIP-Spec] 3.3 and 9.1.3.2.12)
 - 998 i. Certificate
 - 999 ii. Public Key
 - 1000 iii. Private Key
 - 1001 b. Cryptographic Algorithm ([KMIP-Spec] 3.4 and 9.1.3.2.13)
 - 1002 i. RSA
 - 1003 c. Certificate Type ([KMIP-Spec] 3.8 and 9.1.3.2.6)
 - 1004 i. X.509
 - 1005 d. X.509 Certificate Identifier ([KMIP-Spec] 3.10)
 - 1006 e. X.509 Certificate Subject ([KMIP-Spec] 3.11)
 - 1007 f. X.509 Certificate Issuer ([KMIP-Spec] 3.12)
 - 1008 g. Link ([KMIP-Spec] 3.35 and 9.1.3.2.20)
 - 1009 i. Certificate Link
 - 1010 ii. Public Key Link

- 1011 iii. Private Key Link
- 1012 iv. Replacement Object Link
- 1013 v. Replaced Object Link
- 1014 h. Certificate Request Type ([KMIP-Spec] 4.7, 4.8 and 9.1.3.2.22)
- 1015 i. PKCS#10
- 1016 ii. PEM
- 1017 5. Supports the following subsets of enumerated objects:
- 1018 a. Key Format Type ([KMIP-Spec] 2.1.3 and 9.1.3.2.3)
- 1019 i. PKCS#1
- 1020 ii. X.509
- 1021 6. Optionally supports any clause within [KMIP-Spec] that is not listed above
- 1022 7. Optionally supports extensions outside the scope of this standard (e.g., vendor extensions,
- 1023 Conformance Clauses) that do not contradict any requirements within this standard

1024 **5.20 Asymmetric Key Foundry and Certificate Client Conformance**
 1025 **Clauses**

1026 This proposal intends to meet this OASIS requirement by building on the KMIP Conformance Clauses to
 1027 request basic asymmetric key services for central key generation (by the key server). The intent is to
 1028 simply allow key and certificate creation and serving with very limited key types.

1029 **5.20.1 Implementation Conformance**

1030 An implementation is a conforming KMIP Asymmetric Key Foundry and Certificate Client if the
 1031 implementation meets the conditions as outlined in the following section.

1032 **5.20.2 Conformance as a Basic Asymmetric Key Foundry and Certificate**
 1033 **Client**

1034 An implementation conforms to this specification as a KMIP Asymmetric Key Foundry and Certificate
 1035 Client (Central Generation) if it meets the following conditions:

- 1036 1. Supports the conditions required by the KMIP Client conformance clauses ([KMIP-Spec] 12.2)
- 1037 and Baseline Client conformance clause ([KMIP-Prof] 5.12)
- 1038 2. Supports the Baseline KMIP Client Profile (KMIP-Prof 4.2)
- 1039 3. Supports the following additional objects:
- 1040 a. Certificate ([KMIP-Spec] 2.2.1)
- 1041 b. Public Key ([KMIP-Spec] 2.2.3)
- 1042 c. Private Key ([KMIP-Spec] 2.2.4)
- 1043 4. Supports the following client-to-server operations:
- 1044 a. Create Key Pair ([KMIP-Spec] 4.2)
- 1045 b. Re-key Key Pair ([KMIP-Spec] 4.5)
- 1046 c. Certify ([KMIP-Spec] 4.7)
- 1047 a. Re-Certify ([KMIP-Spec] 4.8)
- 1048 5. Supports the following subset of enumerated attributes:
- 1049 a. Object Type ([KMIP-Spec] 3.3 and 9.1.3.2.12)
- 1050 i. Certificate
- 1051 ii. Public Key

- 1052 iii. Private Key
- 1053 b. Cryptographic Algorithm ([KMIP-Spec] 3.4 and 9.1.3.2.13)
- 1054 i. RSA
- 1055 c. Certificate Type ([KMIP-Spec] 3.8 and 9.1.3.2.6)
- 1056 i. X.509
- 1057 d. X.509 Certificate Identifier ([KMIP-Spec] 3.10)
- 1058 e. X.509 Certificate Subject ([KMIP-Spec] 3.11)
- 1059 f. X.509 Certificate Issuer ([KMIP-Spec] 3.12)
- 1060 g. Link ([KMIP-Spec] 3.35 and 9.1.3.2.20)
- 1061 i. Certificate Link
- 1062 ii. Public Key Link
- 1063 iii. Private Key Link
- 1064 iv. Replacement Object Link
- 1065 v. Replaced Object Link
- 1066 h. Certificate Request Type ([KMIP-Spec] 4.7, 4.8 and 9.1.3.2.22)
- 1067 i. PKCS#10
- 1068 ii. PEM
- 1069 6. Supports the following subset of enumerated objects:
- 1070 a. Key Format Type ([KMIP-Spec] 2.1.3 and 9.1.3.2.3)
- 1071 i. PKCS#1
- 1072 ii. X.509
- 1073 iii. Transparent RSA private key ([KMIP-Spec] 2.1.7.4)
- 1074 iv. Transparent RSA public key ([KMIP-Spec] 2.1.7.4)
- 1075 7. Optionally supports any clause within [KMIP-Spec] that is not listed above
- 1076 8. Optionally supports extensions outside the scope of this standard (e.g., vendor extensions,
- 1077 Conformance Clauses) that do not contradict any requirements within this standard
- 1078

1079 **5.21 Storage Client Conformance Clauses**

1080 This proposal intends to meet this OASIS requirement by building on the KMIP Client Conformance
1081 Clauses to request services for storage-related capabilities by the key server...

1082 **5.21.1 Implementation Conformance**

1083 An implementation is a conforming KMIP Storage Client if the implementation meets the conditions as
1084 outlined in the following section.

1085 **5.21.2 Conformance as a Storage Client**

1086 An implementation conforms to this specification as a KMIP Storage Client if it meets the following
1087 conditions:

- 1088 1. Supports the conditions required by the KMIP Client conformance clauses ([KMIP-Spec] 12.2)
- 1089 2. Supports the Baseline Client Conformance Clause (Section 5.12)
- 1090 3. Supports the Symmetric Key Store Client Conformance Clause (Section 5.14)
- 1091 4. Supports the Symmetric Key Foundry Client Conformance Clause (Section 5.15)
- 1092 5. Optionally supports any clause within [KMIP-Spec] that is not listed above

1093 6. Optionally supports extensions outside the scope of this standard (e.g., vendor extensions,
1094 Conformance Clauses) that do not contradict any requirements within this standard
1095
1096
1097

1098

Appendix A. Acknowledgements

1099 The following individuals have participated in the creation of this specification and are gratefully
1100 acknowledged:

1101 **Original Authors of the initial contribution:**

1102 Bruce Rich, IBM
1103 Subhash Sankuratripati, NetApp
1104

1105 **Participants:**

1106
1107 Hal Aldridge, Sypris Electronics
1108 Mike Allen, Symantec
1109 Gordon Arnold, IBM
1110 Todd Arnold, IBM
1111 Matthew Ball, Oracle Corporation
1112 Elaine Barker, NIST
1113 Peter Bartok, Venafi, Inc.
1114 Mathias Björkqvist, IBM
1115 Kelley Burgin, National Security Agency
1116 John Clark, Hewlett-Packard
1117 Tom Clifford, Symantec Corp.
1118 Graydon Dodson, Lexmark International Inc.
1119 Chris Dunn, SafeNet, Inc.
1120 Michael Duren, Sypris Electronics
1121 Paul Earsy, SafeNet, Inc.
1122 Stan Feather, Hewlett-Packard
1123 Indra Fitzgerald, Hewlett-Packard
1124 Alan Frindell, SafeNet, Inc.
1125 Judith Furlong, EMC Corporation
1126 Jonathan Geater, Thales e-Security
1127 Susan Gleeson, Oracle
1128 Robert Griffin, EMC Corporation
1129 Paul Grojean, Individual
1130 Robert Haas, IBM
1131 Thomas Hardjono, M.I.T.
1132 Steve He, Vormetric Inc.
1133 Kurt Heberlein, Hewlett-Packard
1134 Joel Hockey, Cryptsoft Pty Ltd.
1135 Larry Hofer, Emulex Corporation
1136 Brandon Hoff, Emulex Corporation
1137 Walt Hubis, NetApp
1138 Tim Hudson, Cryptsoft Pty Ltd.
1139 Jay Jacobs, Target Corporation
1140 Glen Jaquette, IBM
1141 Scott Kipp, Brocade Communications Systems, Inc.
1142 Kathy Kriese, Symantec Corporation
1143 David Lawson, Emulex Corporation
1144 John Leiseboer, Quintessence Labs
1145 Hal Lockhart, Oracle Corporation
1146 Robert Lockhart, Thales e-Security
1147 Anne Luk, Cryptsoft Pty Ltd.
1148 Shyam Mankala, EMC Corporation
1149 Upendra Mardikar, PayPal Inc.

1150 Luther Martin, Voltage Security
1151 Hyrum Mills, Mitre Corporation
1152 Bob Nixon, Emulex Corporation
1153 René Pawlitzek, IBM
1154 John Peck, IBM
1155 Rob Philpott, EMC Corporation
1156 Denis Pochuev, SafeNet, Inc.
1157 Ajai Puri, SafeNet Inc.
1158 Peter Reed, SafeNet Inc.
1159 Bruce Rich, IBM
1160 Warren Robbins, Credant Systems
1161 Saikat Saha, SafeNet, Inc.
1162 Subhash Sankuratripati, NetApp
1163 Mark Schiller, Hewlett-Packard
1164 Brian Spector, Certivox
1165 Terence Spies, Voltage Security
1166 Marcus Streets, Thales e-Security
1167 Kiran Thota, VMware
1168 Sean Turner, IECA, Inc.
1169 Paul Turner, Venafi, Inc.
1170 Marko Vukolić, EURECOM
1171 Rod Wideman, Quantum Corporation
1172 Steven Wierenga, Hewlett-Packard
1173 Peter Yee, EMC Corporation
1174 Krishna Yellepeddy, IBM
1175 Michael Yoder, Voremetric. Inc.
1176 Peter Zelechowski, Election Systems & Software
1177 Magda Zdunkiewicz, Cryptsoft

Appendix B. Revision History

Revision	Date	Editor	Changes Made
wd 01	2011-05-18	Robert Griffin	Initial revision of KMIP V1.0 Profiles committee draft to include profile test case specifications.
wd 02	2011-07-14	Robert Griffin	Update to include draft client profiles
wd 03	2011-08-2	Robert Griffin	Update to include baseline profiles
wd 04	2011-09-9	Robert Griffin	Update to include storage client profile
wd 05	2011-10-06	Robert Griffin	Update to include required authentication, reformat profiles and clauses, and to remove test scenarios
wd 06	2011-10-19	Robert Griffin	Reformatted in OASIS standards track document format.
wd 07	2011-12-01	Robert Griffin	Incorporates revisions from "KMIP Profiles Conformance Proposal rev 29oct2011" and minor edits...
wd 08	2011-12-17	Robert Griffin	Incorporates editorial corrections.
wd 09	2011-12-20	Robert Griffin	References corrected
CND	2012-1-4	OASIS admin	Committee Specification Draft for public review
wd 10	2012-4-4	Robert Griffin	Comments from public review incorporated.
wd 11	2012-4-26	Robert Griffin	Updated contributors list.