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

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

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.

[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.

2 Profiles

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

2.1 Guidelines for Specifying Conformance Clauses

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

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

2.2 Guidelines for Specifying Authentication Suites

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

2.3 Guidelines for Specifying KMIP Profiles

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

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

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

2.4 Guidelines for Validating Conformance to KMIP Profiles

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

- All objects specified as required in that profile
- All operations specified as required in that profile
- All attributes specified as required in that profile
- The defined wire protocols (TLS, SSL, IPSec, etc…) for that profile
- The defined methods of authentication for that profile

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

- All objects specified as required in that profile
- All operations specified as required in that profile
- All attributes specified as required in that profile
- The defined wire protocols (TLS, SSL, IPSec, etc…) for that profile
- The defined methods of authentication for that profile

3 Authentication Suites

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

3.1 Basic Authentication Suite

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

3.1.1 Protocols

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

3.1.2 Cipher Suites

Conformant KMIP servers SHALL support the following cipher suites:

- `TLS_RSA_WITH_AES_128_CBC_SHA`

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

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

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

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

3.1.3 Client Authenticity

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

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

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

3.1.4 Object Owner

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

3.1.5 KMIP Port Number

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

3.2 TLS 1.2 Authentication Suite

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

3.2.1 Protocols

Conformant KMIP servers SHALL support TLSv1.2

3.2.2 Cipher Suites

Conformant KMIP servers SHALL support the following cipher suites:

- `TLS_RSA_WITH_AES_256_CBC_SHA256`
- `TLS_RSA_WITH_AES_128_CBC_SHA256`

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

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

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

3.2.3 Client Authenticity

Same as the basic authentication suite Section 3.1.3.

3.2.4 Object Owner

Same as the basic authentication suite Section 3.1.4.

3.2.5 KMIP Port Number

Same as the basic authentication suite Section 3.1.5.

4 KMIP Profiles

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

4.1 Basic Discover Versions Server Profile

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

4.2 Basic Baseline Server KMIP Profile

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

4.3 Basic Secret Data Server KMIP Profile

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

4.4 Basic Symmetric Key Store and Server KMIP Profile

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

4.5 Basic Symmetric Key Foundry and Server KMIP Profile

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

4.6 Basic Asymmetric Key Store Server KMIP Profile

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

4.7 Basic Asymmetric Key and Certificate Store Server KMIP Profile

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

4.8 Basic Asymmetric Key Foundry and Server KMIP Profile

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

4.9 Basic Certificate Server KMIP Profile

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

4.10 Basic Asymmetric Key Foundry and Certificate Server KMIP Profile

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

4.11 Discover Versions TLS 1.2 Authentication Server Profile

A profile that consists of the tuple {Discover Versions Server Conformance Clause, TLS 1.2 Authentication Suite}

4.12 Baseline Server TLS 1.2 Authentication KMIP Profile

A profile that consists of the tuple {Baseline Server Conformance Clause, TLS 1.2 Authentication Suite}

4.13 Secret Data Server TLS 1.2 Authentication KMIP Profile

A profile that consists of the tuple {Secret Data Server Conformance Clause, TLS 1.2 Authentication Suite}

4.14 Symmetric Key Store and Server TLS 1.2 Authentication KMIP Profile

A profile that consists of the tuple {Basic Symmetric Key Store and Server Conformance Clause, TLS 1.2 Authentication Suite}

4.15 Symmetric Key Foundry and Server TLS 1.2 Authentication KMIP Profile

A profile that consists of the tuple {Basic Symmetric Key Foundry and Server Conformance Clause, TLS

4.16 Asymmetric Key Store Server TLS 1.2 Authentication KMIP Profile

A profile that consists of the tuple {Asymmetric Key Store Server Conformance Clause, TLS 1.2 Authentication Suite)

4.17 Asymmetric Key and Certificate Store Server TLS 1.2 Authentication KMIP Profile

A profile that consists of the tuple {Asymmetric Key Foundry and Certificate Store Server Conformance Clause, TLS 1.2 Authentication Suite)

4.18 Asymmetric Key Foundry and Server TLS 1.2 Authentication KMIP Profile

A profile that consists of the tuple {Asymmetric Key Foundry and Server Conformance Clause, TLS 1.2 Authentication Suite)

4.19 Certificate Server TLS 1.2 Authentication KMIP Profile

A profile that consists of the tuple {Certificate Server Conformance Clause, TLS 1.2 Authentication Suite}

4.20 Asymmetric Key Foundry and Certificate Server TLS 1.2 Authentication KMIP Profile

A profile that consists of the tuple {Asymmetric Key Foundry and Certificate Store Server Conformance Clause, TLS 1.2 Authentication Suite)

4.21 Basic Discover Versions Client KMIP Profile

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

4.22 Basic Baseline Client KMIP Profile

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

4.23 Basic Secret Data Client KMIP Profile

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

4.24 Basic Symmetric Key Store Client KMIP Profile

A profile that consists of the tuple {Basic Symmetric Key Store Client Conformance Clause, Basic Authentication Suite}

4.25 Basic Symmetric Key Foundry Client KMIP Profile

A profile that consists of the tuple {Basic Symmetric Key Foundry Client Conformance Clause, Basic Authentication Suite}

4.26 Basic Asymmetric Key Store Client KMIP Profile

A profile that consists of the tuple {Basic Asymmetric Key Store Client Conformance Clause, Basic Authentication Suite}

4.27 Basic Asymmetric Key and Certificate Store Client KMIP Profile

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

4.28 Basic Asymmetric Key Foundry Client KMIP Profile

A profile that consists of the tuple {Basic Asymmetric Key Foundry Client Conformance Clause, Basic Authentication Suite}

4.29 Basic Certificate Client KMIP Profile

A profile that consists of the tuple {Basic Certificate Client Conformance Clause, Basic Authentication Suite}

4.30 Basic Asymmetric Key Foundry and Certificate Client KMIP Profile

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

4.31 Discover Versions Client TLS 1.2 Authentication KMIP Profile

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

4.32 Baseline Client TLS 1.2 Authentication KMIP Profile

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

4.33 Secret Data Client TLS 1.2 Authentication KMIP Profile

A profile that consists of the tuple {Secret Data Client Conformance Clause, TLS 1.2 Authentication Suite}

4.34 Symmetric Key Store Client TLS 1.2 Authentication KMIP Profile

A profile that consists of the tuple {Basic Symmetric Key Store Client Conformance Clause, TLS 1.2 Authentication Suite}

4.35 Symmetric Key Foundry Client TLS 1.2 Authentication KMIP Profile

A profile that consists of the tuple {Basic Symmetric Key Foundry and Server Conformance Clause, TLS 1.2 Authentication Suite}

4.36 Asymmetric Key Store Client TLS 1.2 Authentication KMIP Profile

A profile that consists of the tuple {Basic Asymmetric Key Store Client Conformance Clause, TLS 1.2 Authentication Suite}

4.37 Asymmetric Key and Certificate Store Client TLS 1.2 Authentication KMIP Profile

A profile that consists of the tuple {Basic Asymmetric Key and Certificate Store Client Conformance Clause, TLS 1.2 Authentication Suite}

4.38 Asymmetric Key Foundry Client TLS 1.2 Authentication KMIP Profile

A profile that consists of the tuple {Basic Asymmetric Key Foundry Client Conformance Clause, TLS 1.2 Authentication Suite}

4.39 Certificate Client TLS 1.2 Authentication KMIP Profile

A profile that consists of the tuple {Basic Certificate Client Conformance Clause, TLS 1.2 Authentication Suite}

4.40 Asymmetric Key Foundry and Certificate Client TLS 1.2 Authentication KMIP Profile

A profile that consists of the tuple {Basic Asymmetric Key Foundry and Certificate Client Conformance Clause, TLS 1.2 Authentication Suite}

4.41 Storage Client KMIP Profile

A profile that consists of the tuple {Storage Client Conformance Clause, Basic Authentication Suite}

4.42 Storage Client TLS 1.2 Authentication KMIP Profile

A profile that consists of the tuple {Storage Client Conformance Clause, TLS 1.2 Authentication Suite}

5 Conformance Clauses

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

5.1 Discover Versions Server Clause

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

5.1.1. Implementation Conformance

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

5.1.2 Conformance of a Discover Versions Server

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

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

5.2 Baseline Server Clause

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

5.2.1 Implementation Conformance

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

5.2.2 Conformance of a KMIP Baseline Server

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

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

i. Cryptographic Usage Mask ([KMIP-Spec] 3.19)
j. State ([KMIP-Spec] 3.22)
k. Initial Date ([KMIP-Spec] 3.23)
l. Activation Date ([KMIP-Spec] 3.24)
m. Deactivation Date ([KMIP-Spec] 3.27)
n. Compromise Occurrence Date ([KMIP-Spec] 3.29)
o. Compromise Date ([KMIP-Spec] 3.30)
p. Revocation Reason ([KMIP-Spec] 3.31)
q. Last Change Date ([KMIP-Spec] 3.38)

4. Supports the ID Placeholder ([KMIP-Spec] 4)
5. Supports the following client-to-server operations:
   a. Locate ([KMIP-Spec] 4.9)
   b. Check ([KMIP-Spec] 4.10)
   c. Get ([KMIP-Spec] 4.11)
   d. Get Attributes ([KMIP-Spec] 4.12)
   e. Get Attribute List ([KMIP-Spec] 4.13)
   f. Add Attribute ([KMIP-Spec] 4.14)
   g. Modify Attribute ([KMIP-Spec] 4.15)
   h. Delete Attribute ([KMIP-Spec] 4.16)
   i. Activate ([KMIP-Spec] 4.19)
   j. Revoke ([KMIP-Spec] 4.20)
   k. Destroy ([KMIP-Spec] 4.21)
   l. Query ([KMIP-Spec] 4.25)
   m. Discover Versions ([KMIP-Spec] 4.26)
6. Supports the following message contents:
   a. Protocol Version ([KMIP-Spec] 6.1)
   b. Operation ([KMIP-Spec] 6.2)
   c. Maximum Response Size ([KMIP-Spec] 6.3)
   d. Unique Batch Item ID ([KMIP-Spec] 6.4)
   e. Time Stamp ([KMIP-Spec] 6.5)
   f. Asynchronous Indicator ([KMIP-Spec] 6.7)
   g. Result Status ([KMIP-Spec] 6.9)
   h. Result Reason ([KMIP-Spec] 6.10)
   i. Batch Order Option ([KMIP-Spec] 6.12)
   j. Batch Error Continuation Option ([KMIP-Spec] 6.13)
   k. Batch Count ([KMIP-Spec] 6.14)
   l. Batch Item ([KMIP-Spec] 6.15)
7. Supports Message Format ([KMIP-Spec] 7)
8. Supports Authentication ([KMIP-Spec] 8)
9. Supports the TTLV encoding ([KMIP-Spec] 9.1)
10. Supports the transport requirements ([KMIP-Spec] 10)
11. Supports Error Handling ([KMIP-Spec] 11) for any supported object, attribute, or operation
12. Optionally supports any clause within [KMIP-Spec] that is not listed above
13. Optionally supports extensions outside the scope of this standard (e.g., vendor extensions, conformance clauses) that do not contradict any KMIP requirements

5.3 Secret Data Server Clause

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

5.3.1 Implementation Conformance

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

5.3.2 Conformance of a Secret Data Server

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

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

5.4 Symmetric Key Store and Server Conformance Clause

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

5.4.1 Implementation Conformance

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

5.4.2 Conformance as a Symmetric Key Store and Server

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

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

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

5.5 Symmetric Key Foundry and Server Conformance Clause

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

5.5.1 Implementation Conformance

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

5.5.2 Conformance as a KMIP Symmetric Key Foundry and Server

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

1. Supports the conditions required by the KMIP Server conformance clauses. ([KMIP-Spec] 12.1) and Baseline Server conformance clause ([KMIP-Prof] 5.2)
2. Supports the following additional objects
    a. Symmetric Key ([KMIP-Spec] 2.2.2)
3. Supports the following client-to-server operations:
    a. Create ([KMIP-Spec] 4.1)
4. Supports the following attributes:
    a. Process Start Date ([KMIP-Spec] 3.25)
    b. Protect Stop Date ([KMIP-Spec] 3.26)
5. Supports the following subsets of enumerated attributes:
    a. Cryptographic Algorithm ([KMIP-Spec] 3.4 and 9.1.3.2.13)
        i. 3DES
        ii. AES
    b. Object Type ([KMIP-Spec] 3.3 and 9.1.3.2.12)
        i. Symmetric Key
6. Supports the following subsets of enumerated objects:
    a. Key Format Type ([KMIP-Spec] 2.1.3 and 9.1.3.2.3)
        i. Raw

ii. Transparent Symmetric Key

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

5.6 Asymmetric Key Store Server Conformance Clauses

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

5.6.1 Implementation Conformance

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

5.6.2 Conformance as an Asymmetric Key Store Server

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

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

5.7 Asymmetric Key and Certificate Store Server Conformance Clauses

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

5.7.1 Implementation Conformance

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

5.7.2 Conformance as a Asymmetric Key and Certificate Store Server

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

1. Supports the conditions required by the KMIP Server conformance clauses ([KMIP-Spec] 12.1) and Baseline Server conformance clause ([KMIP-Prof] 5.2)
2. Supports the following subsets of additional objects:
    a. Certificate ([KMIP-Spec] 2.2.1)
    b. Public Key ([KMIP-Spec] 2.2.3)
    c. Private Key ([KMIP-Spec] 2.2.4)
3. Supports the following client-to-server operations:
    a. Register ([KMIP-Spec] 4.3)
4. Supports the following subset of enumerated attributes:
    a. Object Type ([KMIP-Spec] 3.3 and 9.1.3.2.12)
        i. Certificate
        ii. Public Key
        iii. Private Key
    b. Cryptographic Algorithm ([KMIP-Spec] 3.4 and 9.1.3.2.13)
        i. RSA
    c. Certificate Type ([KMIP-Spec] 3.8 and 9.1.3.2.6)
        i. X.509
    d. X.509 Certificate Identifier ([KMIP-Spec] 3.10)
    e. X.509 Certificate Subject ([KMIP-Spec] 3.11)
    f. X.509 Certificate Issuer ([KMIP-Spec] 3.12)
    g. Link ([KMIP-Spec] 3.35 and 9.1.3.2.20)
        a. Certificate Link
        b. Public Key Link
        c. Private Key Link
5. Supports the following subset of enumerated objects:
    d. Key Format Type ([KMIP-Spec] 2.1.3 and 9.1.3.2.3)
        i. PKCS#1
        ii. X.509
6. Optionally supports any clause within [KMIP-Spec] that is not listed above

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

5.8 Asymmetric Key Foundry and Server Conformance Clauses

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

5.8.1 Implementation Conformance

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

5.8.2 Conformance as a Asymmetric Key Foundry and Server

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

1. Supports the conditions required by the KMIP Server conformance clauses ([KMIP-Spec] 12.1) and Baseline Server conformance clause ([KMIP-Prof] 5.2)
2. Supports the following additional objects:
    a. Public Key ([KMIP-Spec] 2.2.3)
    b. Private Key ([KMIP-Spec] 2.2.4)
3. Supports the following client-to-server operations:
    a. Create Key Pair ([KMIP-Spec] 4.2)
    b. Re-key Key Pair ([KMIP-Spec] 4.5)
4. Supports the following subset of enumerated attributes:
    a. Object Type ([KMIP-Spec] 3.3 and 9.1.3.2.12)
        i. Public Key
        ii. Private Key
    b. Cryptographic Algorithm ([KMIP-Spec] 3.4 and 9.1.3.2.13)
        i. RSA
    c. Link ([KMIP-Spec] 3.35 and 9.1.3.2.20)
        i. Public Key Link
        ii. Private Key Link
        iii. Replacement Object Link
        iv. Replaced Object Link
5. Supports the following subset of enumerated objects:
    c. Key Format Type ([KMIP-Spec] 2.1.3 and 9.1.3.2.3)
        i. PKCS#1
        ii. Transparent RSA private key ([KMIP-Spec] 2.1.7.4)
        iii. Transparent RSA public key ([KMIP-Spec] 2.1.7.5)
6. Optionally supports any clause within [KMIP-Spec] that is not listed above
7. Optionally supports extensions outside the scope of this standard (e.g., vendor extensions, Conformance Clauses) that do not contradict any requirements within this standard

5.9 Certificate Server Conformance Clauses

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

5.9.1 Implementation Conformance

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

5.9.2 Conformance as a Certificate Server

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

1. Supports the conditions required by the KMIP Server conformance clauses ([KMIP-Spec] 12.1) and Baseline Server conformance clause ([KMIP-Prof] 5.2)
2. Supports the following additional objects:
    a. Certificate ([KMIP-Spec] 2.2.1)
    b. Public Key ([KMIP-Spec] 2.2.3)
    c. Private Key ([KMIP-Spec] 2.2.4)
3. Supports the following client-to-server operations:
    a. Certify ([KMIP-Spec] 4.7)
    b. Re-Certify ([KMIP-Spec] 4.8)
4. Supports the following subset of enumerated attributes:
    a. Object Type ([KMIP-Spec] 3.3 and 9.1.3.2.12)
        i. Certificate
        ii. Public Key
        iii. Private Key
    b. Cryptographic Algorithm ([KMIP-Spec] 3.4 and 9.1.3.2.13)
        i. RSA
    c. Certificate Type ([KMIP-Spec] 3.8 and 9.1.3.2.6)
        i. X.509
    d. X.509 Certificate Identifier ([KMIP-Spec] 3.10)
    e. X.509 Certificate Subject ([KMIP-Spec] 3.11)
    f. X.509 Certificate Issuer ([KMIP-Spec] 3.12)
    g. Link ([KMIP-Spec] 3.35 and 9.1.3.2.20)
        i. Certificate Link
        ii. Public Key Link
        iii. Private Key Link
        iv. Replacement Object Link
        v. Replaced Object Link
    h. Certificate Request Type ([KMIP-Spec] 4.7, 4.8 and 9.1.3.2.22)
        i. PKCS#10
        ii. PEM
5. Supports the following subsets of enumerated objects:
    a. Key Format Type ([KMIP-Spec] 2.1.3 and 9.1.3.2.3)

i. PKCS#1
ii. X.509

6. Optionally supports any clause within [KMIP-Spec] that is not listed above
7. Optionally supports extensions outside the scope of this standard (e.g., vendor extensions, Conformance Clauses) that do not contradict any requirements within this standard

5.10 Asymmetric Key Foundry and Certificate Server Conformance Clauses

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

5.10.1 Implementation Conformance

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

5.10.2 Conformance as a Asymmetric Key Foundry and Certificate Server

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

1. Supports the conditions required by the KMIP Server conformance clauses ([KMIP-Spec] 12.1) and Baseline Server conformance clause ([KMIP-Prof] 5.2)
2. Supports the following additional objects:
   a. Certificate ([KMIP-Spec] 2.2.1)
   b. Public Key ([KMIP-Spec] 2.2.3)
   c. Private Key ([KMIP-Spec] 2.2.4)
3. Supports the following client-to-server operations:
   a. Create Key Pair ([KMIP-Spec] 4.2)
   b. Re-key Key Pair ([KMIP-Spec] 4.5)
   c. Certify ([KMIP-Spec] 4.7)
   d. Re-Certify ([KMIP-Spec] 4.8)
4. Supports the following subset of enumerated attributes:
   a. Object Type ([KMIP-Spec] 3.3 and 9.1.3.2.12)
      i. Certificate
      ii. Public Key
      iii. Private Key
   b. Cryptographic Algorithm ([KMIP-Spec] 3.4 and 9.1.3.2.13)
      i. RSA
   c. Certificate Type ([KMIP-Spec] 3.8 and 9.1.3.2.6)
      i. X.509
   d. X.509 Certificate Identifier ([KMIP-Spec] 3.10)
   e. X.509 Certificate Subject ([KMIP-Spec] 3.11)
   f. X.509 Certificate Issuer ([KMIP-Spec] 3.12)
   g. Link ([KMIP-Spec] 3.35 and 9.1.3.2.20)
      i. Certificate Link

ii. Public Key Link
iii. Private Key Link
iv. Replacement Object Link
v. Replaced Object Link

h. Certificate Request Type ([KMIP-Spec] 4.7, 4.8 and 9.1.3.2.22)
i. PKCS#10
ii. PEM

5. Supports the following subset of enumerated objects:
   d. Key Format Type ([KMIP-Spec] 2.1.3 and 9.1.3.2.3)
      i. PKCS#1
      ii. X.509
      iii. Transparent RSA private key ([KMIP-Spec] 2.1.7.4)
      iv. Transparent RSA public key ([KMIP-Spec] 2.1.7.4)
6. Optionally supports any clause within [KMIP-Spec] that is not listed above
7. Optionally supports extensions outside the scope of this standard (e.g., vendor extensions, Conformance Clauses) that do not contradict any requirements within this standard

5.11 Discover Versions Client Clause

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

5.11.1 Implementation Conformance

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

5.11.2 Conformance of a Discover Versions Client

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

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

5.12 Baseline Client Clause

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

5.12.1 Implementation Conformance

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

5.12.2 Conformance of a KMIP Baseline Client

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

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

a. Attribute ([KMIP-Spec] 2.1.1)
b. Credential ([KMIP-Spec] 2.1.2)
c. Key Block ([KMIP-Spec] 2.1.3)
d. Key Value ([KMIP-Spec] 2.1.4)
e. Template-Attribute Structure ([KMIP-Spec] 2.1.8)

3. Supports the following subsets of attributes:
    a. Unique Identifier ([KMIP-Spec] 3.1)
    b. Name ([KMIP-Spec] 3.2)
    c. Object Type ([KMIP-Spec] 3.3)
    d. Cryptographic Algorithm ([KMIP-Spec] 3.4)
    e. Cryptographic Length ([KMIP-Spec] 3.5)
    f. Cryptographic Parameters ([KMIP-Spec] 3.6)
    g. Digest ([KMIP-Spec] 3.17)
    h. Default Operation Policy ([KMIP-Spec] 3.18.2)
    i. Cryptographic Usage Mask ([KMIP-Spec] 3.19)
    j. State ([KMIP-Spec] 3.22)
    k. Initial Date ([KMIP-Spec] 3.23)
    l. Activation Date ([KMIP-Spec] 3.24)
    m. Deactivation Date ([KMIP-Spec] 3.27)
    n. Compromise Occurrence Date ([KMIP-Spec] 3.29)
    o. Compromise Date ([KMIP-Spec] 3.30)
    p. Revocation Reason ([KMIP-Spec] 3.31)
    q. Last Change Date ([KMIP-Spec] 3.38)

4. Supports the ID Placeholder ([KMIP-Spec] 4)
5. Supports the following client-to-server operations:
    a. Locate ([KMIP-Spec] 4.9)
    b. Check ([KMIP-Spec] 4.10)
    c. Get ([KMIP-Spec] 4.11)
    d. Get Attributes ([KMIP-Spec] 4.12)
    e. Get Attribute List ([KMIP-Spec] 4.13)
    f. Add Attribute ([KMIP-Spec] 4.14)
    g. Modify Attribute ([KMIP-Spec] 4.15)
    h. Delete Attribute ([KMIP-Spec] 4.16)
    i. Activate ([KMIP-Spec] 4.19)
    j. Revoke ([KMIP-Spec] 4.20)
    k. Destroy ([KMIP-Spec] 4.21)
    l. Query ([KMIP-Spec] 4.25)
    m. Discover Versions ([KMIP-Spec] 4.26)

6. Supports the following message contents:
    a. Protocol Version ([KMIP-Spec] 6.1)
    b. Operation ([KMIP-Spec] 6.2)
    c. Maximum Response Size ([KMIP-Spec] 6.3)
    d. Unique Batch Item ID ([KMIP-Spec] 6.4)
    e. Time Stamp ([KMIP-Spec] 6.5)
    f. Asynchronous Indicator ([KMIP-Spec] 6.7)
    g. Result Status ([KMIP-Spec] 6.9)
    h. Result Reason ([KMIP-Spec] 6.10)
    i. Batch Order Option ([KMIP-Spec] 6.12)
    j. Batch Error Continuation Option ([KMIP-Spec] 6.13)
    k. Batch Count ([KMIP-Spec] 6.14)
    l. Batch Item ([KMIP-Spec] 6.15)

7. Supports Message Format ([KMIP-Spec] 7)
8. Supports Authentication ([KMIP-Spec] 8)

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

5.13 Secret Data Client Clause

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

5.13.1 Implementation Conformance

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

5.13.2 Conformance of a Secret Data Client

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

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

5.14 Symmetric Key Store Client Conformance Clause

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

5.14.1 Implementation Conformance

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

5.14.2 Conformance as a Symmetric Key Store Client

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

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

5.15 Symmetric Key Foundry Client Conformance Clause

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

5.15.1 Implementation Conformance

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

5.15.2 Conformance as a KMIP Symmetric Key Foundry Client

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

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

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

5.16 Asymmetric Key Store Client Conformance Clauses

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

5.16.1 Implementation Conformance

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

5.16.2 Conformance as a Asymmetric Key Store Client

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

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

i. Public Key Link
ii. Private Key Link

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

5.17 Asymmetric Key and Certificate Store Client Conformance Clauses

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

5.17.1 Implementation Conformance

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

5.17.2 Conformance as an Asymmetric Key and Certificate Store Client

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

1. Supports the conditions required by the KMIP Client conformance clauses ([KMIP-Spec] 12.2) and Baseline Client conformance clause ([KMIP-Prof] 5.12)
2. Supports the following subsets of additional objects:
   a. Certificate ([KMIP-Spec] 2.2.1)
   b. Public Key ([KMIP-Spec] 2.2.3)
   c. Private Key ([KMIP-Spec] 2.2.4)
3. Supports the following client-to-server operations:
   a. Register ([KMIP-Spec] 4.3)
4. Supports the following subset of enumerated attributes:
   a. Object Type ([KMIP-Spec] 3.3 and 9.1.3.2.12)
      i. Certificate
      ii. Public Key
      iii. Private Key
   b. Cryptographic Algorithm ([KMIP-Spec] 3.4 and 9.1.3.2.13)
      i. RSA
   c. Certificate Type ([KMIP-Spec] 3.8 and 9.1.3.2.6)
      i. X.509
   d. X.509 Certificate Identifier ([KMIP-Spec] 3.10)
   e. X.509 Certificate Subject ([KMIP-Spec] 3.11)
   f. X.509 Certificate Issuer ([KMIP-Spec] 3.12)

g. Link ([KMIP-Spec] 3.35 and 9.1.3.2.20)
    a. Certificate Link
    b. Public Key Link
    c. Private Key Link

5. Supports the following subset of enumerated objects:
    a. Key Format Type ([KMIP-Spec] 2.1.3 and 9.1.3.2.3)
        i. PKCS#1
        ii. X.509
6. Optionally supports any clause within [KMIP-Spec] that is not listed above
7. Optionally supports extensions outside the scope of this standard (e.g., vendor extensions, Conformance Clauses) that do not contradict any requirements within this standard

5.18 Asymmetric Key Foundry Client Conformance Clauses

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

5.18.1 Implementation Conformance

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

5.18.2 Conformance as an Asymmetric Key Foundry Client

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

1. Supports the conditions required by the KMIP Server conformance clauses ([KMIP-Spec] 12.2) and Baseline Client conformance clause ([KMIP-Prof] 5.12)
2. Supports the following additional objects:
    a. Public Key ([KMIP-Spec] 2.2.3)
    b. Private Key ([KMIP-Spec] 2.2.4)
3. Supports the following client-to-server operations:
    a. Create Key Pair ([KMIP-Spec] 4.2)
    b. Re-key Key Pair ([KMIP-Spec] 4.5)
4. Supports the following subset of enumerated attributes:
    a. Object Type ([KMIP-Spec] 3.3 and 9.1.3.2.12)
        i. Public Key
        ii. Private Key
    b. Cryptographic Algorithm ([KMIP-Spec] 3.4 and 9.1.3.2.13)
        i. RSA
    c. Link ([KMIP-Spec] 3.35 and 9.1.3.2.20)
        i. Public Key Link
        ii. Private Key Link
        iii. Replacement Object Link
        iv. Replaced Object Link
5. Supports the following subset of enumerated objects:

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

5.19 Certificate Client Conformance Clauses

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

5.19.1 Implementation Conformance

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

5.19.2 Conformance as a Basic Certificate Client

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

1. Supports the conditions required by the KMIP Client conformance clauses ([KMIP-Spec] 12.2) and Baseline Client conformance clause ([KMIP-Prof] 5.12)
2. Supports the following additional objects:
    a. Certificate ([KMIP-Spec] 2.2.1)
    b. Public Key ([KMIP-Spec] 2.2.3)
    c. Private Key ([KMIP-Spec] 2.2.4)
3. Supports the following client-to-server operations:
    a. Certify ([KMIP-Spec] 4.7)
    b. Re-Certify ([KMIP-Spec] 4.8)
4. Supports the following subset of enumerated attributes:
    a. Object Type ([KMIP-Spec] 3.3 and 9.1.3.2.12)
        i. Certificate
        ii. Public Key
        iii. Private Key
    b. Cryptographic Algorithm ([KMIP-Spec] 3.4 and 9.1.3.2.13)
        i. RSA
    c. Certificate Type ([KMIP-Spec] 3.8 and 9.1.3.2.6)
        i. X.509
    d. X.509 Certificate Identifier ([KMIP-Spec] 3.10)
    e. X.509 Certificate Subject ([KMIP-Spec] 3.11)
    f. X.509 Certificate Issuer ([KMIP-Spec] 3.12)
    g. Link ([KMIP-Spec] 3.35 and 9.1.3.2.20)
        i. Certificate Link
        ii. Public Key Link

iii. Private Key Link
iv. Replacement Object Link
v. Replaced Object Link

h. Certificate Request Type ([KMIP-Spec] 4.7, 4.8 and 9.1.3.2.22)
i. PKCS#10
ii. PEM

5. Supports the following subsets of enumerated objects:
   a. Key Format Type ([KMIP-Spec] 2.1.3 and 9.1.3.2.3)
      i. PKCS#1
      ii. X.509
6. Optionally supports any clause within [KMIP-Spec] that is not listed above
7. Optionally supports extensions outside the scope of this standard (e.g., vendor extensions, Conformance Clauses) that do not contradict any requirements within this standard

5.20 Asymmetric Key Foundry and Certificate Client Conformance Clauses

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

5.20.1 Implementation Conformance

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

5.20.2 Conformance as a Basic Asymmetric Key Foundry and Certificate Client

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

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

iii. Private Key
b. Cryptographic Algorithm ([KMIP-Spec] 3.4 and 9.1.3.2.13)
i. RSA
c. Certificate Type ([KMIP-Spec] 3.8 and 9.1.3.2.6)
i. X.509
d. X.509 Certificate Identifier ([KMIP-Spec] 3.10)
e. X.509 Certificate Subject ([KMIP-Spec] 3.11)
f. X.509 Certificate Issuer ([KMIP-Spec] 3.12)
g. Link ([KMIP-Spec] 3.35 and 9.1.3.2.20)
i. Certificate Link
ii. Public Key Link
iii. Private Key Link
iv. Replacement Object Link
v. Replaced Object Link
h. Certificate Request Type ([KMIP-Spec] 4.7, 4.8 and 9.1.3.2.22)
i. PKCS#10
ii. PEM

6. Supports the following subset of enumerated objects:
   a. Key Format Type ([KMIP-Spec] 2.1.3 and 9.1.3.2.3)
      i. PKCS#1
      ii. X.509
      iii. Transparent RSA private key ([KMIP-Spec] 2.1.7.4)
      iv. Transparent RSA public key ([KMIP-Spec] 2.1.7.4)
7. Optionally supports any clause within [KMIP-Spec] that is not listed above
8. Optionally supports extensions outside the scope of this standard (e.g., vendor extensions, Conformance Clauses) that do not contradict any requirements within this standard

5.21 Storage Client Conformance Clauses

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

5.21.1 Implementation Conformance

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

5.21.2 Conformance as a Storage Client

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

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

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

Appendix A. Acknowledgements

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

Original Authors of the initial contribution:

Bruce Rich, IBM
Subhash Sankuratripati, NetApp

Participants:

Hal Aldridge, Sypris Electronics
Mike Allen, Symantec
Gordon Arnold, IBM
Todd Arnold, IBM
Matthew Ball, Oracle Corporation
Elaine Barker, NIST
Peter Bartok, Venafi, Inc.
Mathias Björkqvist, IBM
Kelley Burgin, National Security Agency
John Clark, Hewlett-Packard
Tom Clifford, Symantec Corp.
Graydon Dodson, Lexmark International Inc.
Chris Dunn, SafeNet, Inc.
Michael Duren, Sypris Electronics
Paul Earsy, SafeNet, Inc.
Stan Feather, Hewlett-Packard
Indra Fitzgerald, Hewlett-Packard
Alan Frindell, SafeNet, Inc.
Judith Furlong, EMC Corporation
Jonathan Geater, Thales e-Security
Susan Gleeson, Oracle
Robert Griffin, EMC Corporation
Paul Grojean, Individual
Robert Haas, IBM
Thomas Hardjono, M.I.T.
Steve He, Vormetric Inc.
Kurt Heberlein, Hewlett-Packard
Joel Hockey, Cryptsoft Pty Ltd.
Larry Hofer, Emulex Corporation
Brandon Hoff, Emulex Corporation
Walt Hubis, NetApp
Tim Hudson, Cryptsoft Pty Ltd.
Jay Jacobs, Target Corporation
Glen Jaquette, IBM
Scott Kipp, Brocade Communications Systems, Inc.
Kathy Kriese, Symantec Corporation
David Lawson, Emulex Corporation
John Leiseboer, Quintessence Labs
Hal Lockhart, Oracle Corporation
Robert Lockhart, Thales e-Security
Anne Luk, Cryptsoft Pty Ltd.
Shyam Mankala, EMC Corporation
Upendra Mardikar, PayPal Inc.

Luther Martin, Voltage Security
Hyrum Mills, Mitre Corporation
Bob Nixon, Emulex Corporation
René Pawlitzek, IBM
John Peck, IBM
Rob Philpott, EMC Corporation
Denis Pochuev, SafeNet, Inc.
Ajai Puri, SafeNet Inc.
Peter Reed, SafeNet Inc.
Bruce Rich, IBM
Warren Robbins, Credant Systems
Saikat Saha, SafeNet, Inc.
Subhash Sankuratripati, NetApp
Mark Schiller, Hewlett-Packard
Brian Spector, Certivox
Terence Spies, Voltage Security
Marcus Streets, Thales e-Security
Kiran Thota, VMware
Sean Turner, IECA, Inc.
Paul Turner, Venafi, Inc.
Marko Vukolić, EURECOM
Rod Wideman, Quantum Corporation
Steven Wierenga, Hewlett-Packard
Peter Yee, EMC Corporation
Krishna Yellepeddy, IBM
Michael Yoder, Voremetric. Inc.
Peter Zelechoski, Election Systems & Software
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.