# 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

# 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.

| 45 | **[KMTC]** | *Key Management Interoperability Protocol Test Cases Version 1.1.* 27 July 2012. |
| 46 | | OASIS Committee Note 01. http://docs.oasis- |
| 47 | | open.org/kmip/testcases/v1.1/cn01/kmip-testcases-v1.1-cn01.html. |
| 48 | | |
| 49 | | |
| 50 | | |

# 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

| 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 | | |

# 3 Authentication Suites

94

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

95
96

## 3.1 Basic Authentication Suite

97

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

98
99

### 3.1.1 Protocols

100

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.

101
102
103

### 3.1.2 Cipher Suites

104

Conformant KMIP servers SHALL support the following cipher suites:

105

- `TLS_RSA_WITH_AES_128_CBC_SHA`

106

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`)

107
108
109

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

110

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"

111
112
113

*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.*

114
115
116
117

### 3.1.3 Client Authenticity

118

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

119
120

121

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.

122
123

124

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.

125
126
127
128

### 3.1.4 Object Owner

129

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].

130
131
132
133

### 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)

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


## 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.

### 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)

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

## 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.

### 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.

### 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

## 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

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

## 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.

### 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.

### 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:

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)

| 629 | | | i. | PKCS#1 |

| 630 | | | ii. | X.509 |

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

## 5.10 Asymmetric Key Foundry and Certificate Server Conformance 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.

### 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.

### 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

## 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.

### 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.

### 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)

## 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.

### 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.

### 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 | | |
| 733 | 4. | Supports the ID Placeholder ([KMIP-Spec] 4) |

| | | |
|---|---|---|
| 734 | 5. | Supports the following client-to-server operations: |
| 735 | a. | Locate ([KMIP-Spec] 4.9) |
| 736 | b. | Check ([KMIP-Spec] 4.10) |
| 737 | c. | Get ([KMIP-Spec] 4.11) |
| 738 | d. | Get Attributes ([KMIP-Spec] 4.12) |
| 739 | e. | Get Attribute List ([KMIP-Spec] 4.13) |
| 740 | f. | Add Attribute ([KMIP-Spec] 4.14) |
| 741 | g. | Modify Attribute ([KMIP-Spec] 4.15) |
| 742 | h. | Delete Attribute ([KMIP-Spec] 4.16) |
| 743 | i. | Activate ([KMIP-Spec] 4.19) |
| 744 | j. | Revoke ([KMIP-Spec] 4.20) |
| 745 | k. | Destroy ([KMIP-Spec] 4.21) |
| 746 | l. | Query ([KMIP-Spec] 4.25) |
| 747 | m. | Discover Versions ([KMIP-Spec] 4.26) |
| 748 | | |

| | | |
|---|---|---|
| 749 | 6. | Supports the following message contents: |
| 750 | a. | Protocol Version ([KMIP-Spec] 6.1) |
| 751 | b. | Operation ([KMIP-Spec] 6.2) |
| 752 | c. | Maximum Response Size ([KMIP-Spec] 6.3) |
| 753 | d. | Unique Batch Item ID ([KMIP-Spec] 6.4) |
| 754 | e. | Time Stamp ([KMIP-Spec] 6.5) |
| 755 | f. | Asynchronous Indicator ([KMIP-Spec] 6.7) |
| 756 | g. | Result Status ([KMIP-Spec] 6.9) |
| 757 | h. | Result Reason ([KMIP-Spec] 6.10) |
| 758 | i. | Batch Order Option ([KMIP-Spec] 6.12) |
| 759 | j. | Batch Error Continuation Option ([KMIP-Spec] 6.13) |
| 760 | k. | Batch Count ([KMIP-Spec] 6.14) |
| 761 | l. | Batch Item ([KMIP-Spec] 6.15) |
| 762 | | |
| 763 | 7. | Supports Message Format ([KMIP-Spec] 7) |
| 764 | 8. | Supports Authentication ([KMIP-Spec] 8) |

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

## 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.

### 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.

### 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

## 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.

### 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.

## 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)

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

## 5.17 Asymmetric Key and Certificate Store Client Conformance 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.

### 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.

### 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

## 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.

### 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.

### 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

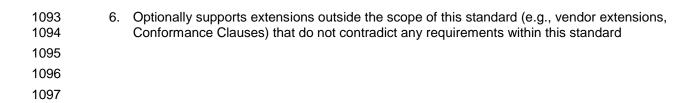964         c.   Link ([KMIP-Spec] 3.35 and 9.1.3.2.20)

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) |

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

## 5.20 Asymmetric Key Foundry and Certificate Client Conformance 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.

### 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.

### 5.20.2 Conformance as a Basic Asymmetric Key Foundry and Certificate 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 |

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

## 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...

### 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.

### 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

# 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.

| 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 Zelechoski, Election Systems & Software |
| 1177 | Magda Zdunkiewicz, Cryptsoft |

# 1178 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. |

1179

1180