OASIS Committee Note

Key Management Interoperability Protocol Test Cases Version 2.0

Committee Note Draft 01

20 December 2018

Specification URIs

This version:

https://docs.oasis-open.org/kmip/kmip-testcases/v2.0/cnd01/kmip-testcases-v2.0-cnd01.docx (Authoritative)

https://docs.oasis-open.org/kmip/kmip-testcases/v2.0/cnd01/kmip-testcases-v2.0-cnd01.html

https://docs.oasis-open.org/kmip/kmip-testcases/v2.0/cnd01/kmip-testcases-v2.0-cnd01.pdf

Previous version:

N/A

Latest version:

https://docs.oasis-open.org/kmip/kmip-testcases/v2.0/kmip-testcases-v2.0.docx (Authoritative)

https://docs.oasis-open.org/kmip/kmip-testcases/v2.0/kmip-testcases-v2.0.html

https://docs.oasis-open.org/kmip/kmip-testcases/v2.0/kmip-testcases-v2.0.pdf

Technical Committee:

OASIS Key Management Interoperability Protocol (KMIP) TC

Chairs:

Tony Cox (tony.cox@cryptsoft.com), Cryptsoft Pty Ltd.

Judith Furlong (Judith.Furlong@dell.com), Dell EMC

Editors:

Tim Hudson (tjh@cryptsoft.com), Cryptsoft Pty Ltd.

Mark Joseph (mark@p6r.com), P6R, Inc

Additional artifacts:

This document is one component of a Work Product that also includes:

- Test cases: https://docs.oasis-open.org/kmip/kmip-testcases/v2.0/cnd01/test-cases/kmip-v2.0/

Related work:

This document replaces or supersedes:

- *Key Management Interoperability Protocol Test Cases Version 1.4.* Edited by Tim Hudson and Mark Joseph. Latest version: http://docs.oasis-open.org/kmip/testcases/v1.4/kmip-testcases-v1.4.html.

This document is related to:

- *Key Management Interoperability Protocol Specification Version 2.0.* Edited by Tony Cox and Charles White. Latest version: https://docs.oasis-open.org/kmip/kmip-spec/v2.0/kmip-spec-v2.0.html.
- *Key Management Interoperability Protocol Profiles Version 2.0.* Edited by Tim Hudson and Robert Lockhart. Latest version: https://docs.oasis-open.org/kmip/kmip-profiles/v2.0/kmip-profiles-v2.0.html.
- *Key Management Interoperability Protocol Usage Guide Version 2.0.* Work in progress.

Abstract:

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

Status:

This is a Non-Standards Track Work Product. The patent provisions of the OASIS IPR Policy do not apply.

This document was last revised or approved by the OASIS Key Management Interoperability Protocol (KMIP) TC on the above date. The level of approval is also listed above. Check the "Latest version" location noted above for possible later revisions of this document.
Technical Committee (TC) members should send comments on this document to the TC's email list. Others should send comments to the TC's public comment list, after subscribing to it by following the instructions at the "Send A Comment" button on the TC's web page at https://www.oasis-open.org/committees/kmip/.

Citation format:

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

[kmip-testcases-v2.0]

Key Management Interoperability Protocol Test Cases Version 2.0. Edited by Tim Hudson and Mark Joseph. 20 December 2018. OASIS Committee Note Draft 01. https://docs.oasis-open.org/kmip/kmip-testcases/v2.0/cnd01/kmip-testcases-v2.0-cnd01.html. Latest version: https://docs.oasis-open.org/kmip/kmip-testcases/v2.0/kmip-testcases-v2.0.html.

Notices

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

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

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

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

Table of Contents

1 Introduction

The purpose of this document is to describe test cases to demonstrate the Key Management Interoperability Protocol (KMIP) [KMIP-SPEC]. The test cases illustrate that the concepts within the protocol are sound and how the protocol may be used when implementing KMIP in applications. These test cases are not intended to fully test an implementation of KMIP.

1.1 References (non-normative)

[KMIP-SPEC]

Key Management Interoperability Protocol Specification Version 2.0. Edited by Tony Cox and Charles White. Latest version: https://docs.oasis-open.org/kmip/kmip-spec/v2.0/kmip-spec-v2.0.html.

[KMIP-PROFILES]

Key Management Interoperability Protocol Profiles Version 2.0. Edited by Tim Hudson and Robert Lockhart. Latest version: https://docs.oasis-open.org/kmip/kmip-profiles/v2.0/kmip-profiles-v2.0.html.

[XML]

XML 1.0 Recommendation, T. Bray, J. Paoli, M. Sperberg-McQueen, Editors, W3C Recommendation, February 10, 1998, http://www.w3.org/TR/1998/REC-xml-19980210. Latest version available at http://www.w3.org/TR/REC-xml.

2 KMIP Test Cases

The test cases define a number of request-response pairs for KMIP operations. Each test case is provided in the XML format specified in [KMIP-PROFILES] intended to be both human-readable and usable by automated tools.

Each test case has a unique label (the section name) which the protocol version as part of the identifier.

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

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

The test cases show one possible way to construct the messages, and the messages shown are not necessarily the only conformant constructions as many items within KMIP are optional and server behavior depends on the server's policy. Support for a test case is predicated on a server matching the test case assumptions and the behavior shown in the request-response pairs.

Symbolic identifiers are of the form $UPPERCASE_NAME followed by optional unique index value. Wherever a symbolic identifier occurs in a test cases the implementation must replace it with a reasonable appearing datum of the expected type. Time values can be specified in terms of an offset from the current time in seconds of the form $NOW or $NOW-n or $NOW+n.

2.1 TC-CERTATTR-1-20

A client registers a certificate and the server creates the certificate attributes based on the subject and issuer distinguished name values.

See test-cases/kmip-v2.0/TC-CERTATTR-1-20.xml

2.2 TC-CREATE-SD-1-20

A client requests a server to create a secret data managed object.

See test-cases/kmip-v2.0/TC-CREATE-SD-1-20.xml

2.3 TC-CS-CORVAL-1-20

A client sets a client correlation value and the server also responds with a server correlation value.

See test-cases/kmip-v2.0/TC-CS-CORVAL-1-20.xml

2.4 TC-DERIVEKEY-1-20

A client uses Derive Key using SHA_256.

See test-cases/kmip-v2.0/TC-DERIVEKEY-1-20.xml

2.5 TC-DERIVEKEY-2-20

A client uses Derive Key using HMAC-SHA_256.

See test-cases/kmip-v2.0/TC-DERIVEKEY-2-20.xml

2.6 TC-DERIVEKEY-3-20

A client uses Derive Key using PBKDF2.

See test-cases/kmip-v2.0/TC-DERIVEKEY-3-20.xml

2.7 TC-DERIVEKEY-4-20

A client uses Derive Key using PBKDF2.

See test-cases/kmip-v2.0/TC-DERIVEKEY-4-20.xml

2.8 TC-DERIVEKEY-5-20

A client uses Derive Key using PBKDF2 and SHA-256.

See test-cases/kmip-v2.0/TC-DERIVEKEY-5-20.xml

2.9 TC-DERIVEKEY-6-20

A client uses Derive Key using ASYMMETRIC_KEY and ECDH.

See test-cases/kmip-v2.0/TC-DERIVEKEY-6-20.xml

2.10 TC-DLOGIN-1-20

Delegated Login.

See test-cases/kmip-v2.0/TC-DLOGIN-1-20.xml

2.11 TC-DLOGIN-2-20

Delegated Login.

See test-cases/kmip-v2.0/TC-DLOGIN-2-20.xml

2.12 TC-ECC-1-20

A client registers and EC private key in ECPrivateKey format and EC public key in X.509 format using the EC cryptographic algorithm.

See test-cases/kmip-v2.0/TC-ECC-1-20.xml

2.13 TC-ECC-2-20

A client registers and EC private key in PKCS8 format and EC public key in X.509 format using the EC cryptographic algorithm.

See test-cases/kmip-v2.0/TC-ECC-2-20.xml

2.14 TC-ECC-3-20

A client registers and EC private key in ECPrivateKey format and EC public key in X.509 format using the EC cryptographic algorithm.

See test-cases/kmip-v2.0/TC-ECC-3-20.xml

2.15 TC-ECDSA-SIGN-1-20

A client registers and EC private key in ECPrivateKey format and EC public key in X.509 format using the EC cryptographic algorithm and performs a Sign operation followed by a Signature Verify operation.

See test-cases/kmip-v2.0/TC-ECDSA-SIGN-1-20.xml

2.16 TC-ECDSA-SIGN-DIGESTEDDATA 1-20

ECDSA Signing with the digested data provided by the client.

See test-cases/kmip-v2.0/TC-ECDSA-SIGN-DIGESTEDDATA-1-20.xml

2.17 TC-EXTRACTABLE-1-20

Show usage of Extractable and Never Extractable

See test-cases/kmip-v2.0/TC-EXTRACTABLE-1-20.xml

2.18 TC-I18N-1-20

Client provides a key name containing a Greek capital Alpha

Note: the encoding in XML has to be correctly converted into the valid UTF-8 format.

See test-cases/kmip-v2.0/TC-I18N-1-20.xml

2.19 TC-I18N-2-20

Client provides a key alternative name containing a Greek capital Alpha

Note: the encoding in XML has to be correctly converted into the valid UTF-8 format.

See test-cases/kmip-v2.0/TC-I18N-2-20.xml

2.20 TC-I18N-3-20

Client provides a customer attribute containing a Greek capital Alpha with the attribute value containing a Greek capital Omega

Note: the encoding in XML has to be correctly converted into the valid UTF-8 format.

See test-cases/kmip-v2.0/TC-I18N-3-20.xml

2.21 TC-IMPEXP-1-20

Import/Export.

See test-cases/kmip-v2.0/TC-IMPEXP-1-20.xml

2.22 TC-IMPEXP-2-20

Import/Export.

See test-cases/kmip-v2.0/TC-IMPEXP-2-20.xml

2.23 TC-IMPEXP-3-20

Import/Export.

See test-cases/kmip-v2.0/TC-IMPEXP-3-20.xml

2.24 TC-LOGIN-1-20

Login.

See test-cases/kmip-v2.0/TC-LOGIN-1-20.xml

2.25 TC-LOGIN-2-20

Login.

See test-cases/kmip-v2.0/TC-LOGIN-2-20.xml

2.26 TC-LOGIN-3-20

Login.

See test-cases/kmip-v2.0/TC-LOGIN-3-20.xml

2.27 TC-MDO-1-20

A client requests a meta-data-only object (no key material).

See test-cases/kmip-v2.0/TC-MDO-1-20.xml

2.28 TC-MDO-2-20

A client requests a meta-data-only object (no key material) and an object with key material and performs Locate that only returns the meta-data-only object.

See test-cases/kmip-v2.0/TC-MDO-2-20.xml

2.29 TC-MDO-3-20

A client requests a meta-data-only object (no key material) using the URL format of the Key Value Location and performs Locate.

See test-cases/kmip-v2.0/TC-MDO-3-20.xml

2.30 TC-OFFSET-1-20

A client requests the server creates a number of symmetric keys and then uses the Offset parameter in Locate to return various items.

See test-cases/kmip-v2.0/TC-OFFSET-1-20.xml

2.31 TC-OFFSET-2-20

A client requests the server creates a number of symmetric keys and then uses the Offset parameter in Locate to return various items.

See test-cases/kmip-v2.0/TC-OFFSET-2-20.xml

2.32 TC-PGP-1-20

Register a PGP public key block and private key block and add appropriate links between the managed objects.

See test-cases/kmip-v2.0/TC-PGP-1-20.xml

2.33 TC-PKCS12-1-20

Register objects and then performs a Get returning in PKCS#12 format

See test-cases/kmip-v2.0/TC-PKCS-12-1-20.xml

2.34 TC-PKCS12-2-20

Register objects in PKCS#12 format and then performs a Get returning the individual objects.

See test-cases/kmip-v2.0/TC-PKCS-12-2-20.xml

2.35 TC-REKEY-1-20

Create a key and perform multiple rekey operations.

See test-cases/kmip-v2.0/TC-REKEY-1-20.xml

2.36 TC-RNG-ATTR-1-20

A client registers a symmetric key including details of the RNG that the client is claiming was used to generate the symmetric key.

See test-cases\kmip-v2.0\TC-RNG-ATTR-1-20.xml

2.37 TC-RNG-ATTR-2-20

A client requests the server creates a symmetric key and it does and also includes the required details of the RNG that was used to generate the symmetric key.

See test-cases\kmip-v2.0\TC-RNG-ATTR-2-20.xml

2.38 TC-RSA-SIGN-DIGESTEDDATA 1-20

RSA Signing with the digested data provided by the client.

See test-cases/kmip-v2.0/TC-RSA-SIGN-DIGESTEDDATA-1-20.xml

2.39 TC-SENSITIVE-1-20

Show usage of Sensitive and Always Sensitive

See test-cases/kmip-v2.0/TC-SENSITIVE-1-20.xml

2.40 TC-SJ-1-20

Create a symmetric key and perform split and join in various combinations.

See test-cases/kmip-v2.0/TC-SJ-1-20.xml

2.41 TC-SJ-2-20

Register a symmetric key and perform split and join in various combinations.

See test-cases/kmip-v2.0/TC-SJ-2-20.xml

2.42 TC-SJ-3-20

Register split keys and perform join in various combinations.

See test-cases/kmip-v2.0/TC-SJ-2-20.xml

2.43 TC-SJ-4-20

Create a symmetric key and perform split and join in various combinations using the XOR method.

See test-cases/kmip-v2.0/TC-SJ-4-20.xml

2.44 TC-SETATTR-1-20

Set Attribute.

See test-cases/kmip-v2.0/TC-SETATTR-1-20.xml

2.45 TC-SETATTR-2-20

Set Attribute.

See test-cases/kmip-v2.0/TC-SETATTR-2-20.xml

2.46 TC-SETATTR-3-20

Set Attribute.

See test-cases/kmip-v2.0/TC-SETATTR-3-20.xml

2.47 TC-STREAM-ENC-1-20

Create a symmetric key and perform encrypt with streaming.

See test-cases/kmip-v2.0/TC-STREAM-ENC-1-20.xml

2.48 TC-STREAM-ENC-2-20

Register a symmetric key and perform encrypt and decrypt with streaming.

See test-cases/kmip-v2.0/TC-STREAM-ENC-2-20.xml

2.49 TC-STREAM-ENCDEC-1-20

Register a symmetric key and perform encrypt with streaming.

See test-cases/kmip-v2.0/TC-STREAM-ENCDEC-1-20.xml

2.50 TC-STREAM-HASH-1-20

Hash operation for data 'abc' in a single request followed immediately by a streaming equivalent for which the result must be identical.

Note: - test vector data from http://csrc.nist.gov/groups/ST/toolkit/documents/Examples/SHA_All.pdf

See test-cases/kmip-v2.0/TC-STREAM-HASH-1-20.xml

2.51 TC-STREAM-HASH-2-20

Hash operation for data 'abc' in a single request followed immediately by a streaming equivalent for which the result must be identical.

Note: - test vector data from http://csrc.nist.gov/groups/ST/toolkit/documents/Examples/SHA_All.pdf

See test-cases/kmip-v2.0/TC-STREAM-HASH-2-20.xml

2.52 TC-STREAM-HASH-3-20

Hash operation for data 'abc' in a single request followed immediately by a streaming equivalent for which the result must be identical.

Note: - test vector data from http://csrc.nist.gov/groups/ST/toolkit/documents/Examples/SHA_All.pdf

See test-cases/kmip-v2.0/TC-STREAM-HASH-3-20.xml

2.53 TC-STREAM-SIGN-1-20

Sign with a known asymmetric key with streaming.

See test-cases/kmip-v2.0/TC-STREAM-SIGN-1-20.xml

2.54 TC-STREAM-SIGNVFY-1-20

Sign and Signature Verify with a known asymmetric key with streaming.

See test-cases/kmip-v2.0/TC-STREAM-SIGNVFY-1-20.xml

2.55 TC-WRAP-1-20

Show usage of Key Wrap Type As Registered.

See test-cases/kmip-v2.0/TC-WRAP-1-20.xml

2.56 TC-WRAP-2-20

Show usage of Key Wrap Type Not Wrapped.

See test-cases/kmip-v2.0/TC-WRAP-2-20.xml

2.57 TC-WRAP-3-20

Show usage of returning wrapped key wrapped with a different wrapping key.

See test-cases/kmip-v2.0/TC-WRAP-3-20.xml

3 KMIP Test Cases Setup

The test cases defined in the previous section all operate independent and assume that the other end of the KMIP connection has been configured to match the assumptions in the test case.

The following scripts allow for setting up the pre-conditions for a number of the test cases and for cleaning up after the test cases have executed – via KMIP operations. A server is not required to use KMIP or to use these scripts for this purpose – they are provided simply because they are useful for some implementations.

Appendix A. Acknowledgments

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

Participants:

[Participant Name, Affiliation | Individual Member]
[Participant Name, Affiliation | Individual Member]

Appendix B. Revision History

Revision	Date	Editor	Changes Made
wd01	20-Dec-2018	Tim Hudson	Initial draft