Key Management Interoperability Protocol Cryptographic Services Version 1.0

Working Draft ~~10~~11

~~14~~ 21 March 2013

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:
Tim Hudson (tjh@cryptsoft.com), Cryptsoft

Related work:
This specification is related to:

- *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.
- *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 Use Cases Version 1.1*. Latest version. http://docs.oasis-open.org/kmip/usecases/v1.1/kmip-usecases-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:
Describes the use of KMIP operations to support cryptographic servers being performed by a KMIP server on behalf of a KMIP client for key management operations.

Status:
This Working Draft (WD) has been produced by one or more TC Members; it has not yet been voted on by the TC or approved as a Committee Draft (Committee Specification Draft or a Committee Note Draft). The OASIS document Approval Process begins officially with a TC vote to approve a WD as a Committee Draft. A TC may approve a Working Draft, revise it, and re-approve it any number of times as a Committee Draft.

Copyright © OASIS Open 2013. All Rights Reserved.

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

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

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

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

OWNERSHIP RIGHTS OR ANY IMPLIED WARRANTIES OF MERCHANTABILITY OR FITNESS FOR A PARTICULAR PURPOSE.

Copyright © OASIS Open 2013. All Rights Reserved.

Table of Contents

Copyright © OASIS Open 2013. All Rights Reserved.

Copyright © OASIS Open 2013. All Rights Reserved.

1 Introduction

For normative definition of the elements of KMIP see the KMIP Specification ([KMIP-SPEC-1_0 and KMIP-SPEC-1_1]) and the KMIP Profiles ([KMIP-PROF]).

Illustrative guidance for the implementation of KMIP clients and servers is provided in the KMIP Usage Guide ([KMIP-UG]) and KMIP Use Cases ([KMIP_UC]).

This document describes the use of KMIP operations to support cryptographic servers being performed by a KMIP server on behalf of a KMIP client for key management operations.

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 [RFC2119].

1.2 Normative References

[RFC1945] T. Berners-Lee, R. Fielding, H. Frystyk, *Hypertext Transfer Protocol -- HTTP/1.0*, http://www.ietf.org/rfc/rfc1945.txt, IETF RFC 1945, May 1996.

[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.

[RFC2246] T. Dierks and C. Allen, *The TLS Protocol, Version 1.0***,** IETF RFC 2246, Jan 1999, http://www.ietf.org/rfc/rfc2246.txt

[KMIP-SPEC-1_0] OASIS Standard, Key Management Interoperability Protocol Specification Version 1.0, October 2010, http://docs.oasis-open.org/kmip/spec/v1.0/os/kmip-spec-1.0-os.doc

[KMIP-SPEC-1_1] *Key Management Interoperability Protocol Specification Version 1.1.* http://docs.oasis-open.org/kmip/spec/v1.1/csd01/kmip-spec-v1.1-csd01.doc Committee Specification Draft 01.1 December 2011.

[KMIP-PROF] *Key Management Interoperability Protocol Usage Guide Version 1.1*. 01 December 2011. OASIS Standard. http://docs.oasis-open.org/kmip/profiles/v1.1/cd01/kmip-profiles-1.1-cd-01.doc

1.3 Non-Normative References

[KMIP-UG] *Key Management Interoperability Protocol Usage Guide Version 1.1.* http://docs.oasis-open.org/kmip/ug/v1.1/kmip-ug-v1.1-cnd01.doc Committee Note Draft, 1 December 2011,

[KMIP-TC] *Key Management Interoperability Protocol Test Cases Version 1.1.* http://docs.oasis-open.org/kmip/usecases/v1.1/kmip-usecases-v1.1-cnd01.doc, Committee Note Draft, 1 December 2011.

Copyright © OASIS Open 2013. All Rights Reserved.

2 Cryptographic Services

The KMIP protocol supports creation and registration of managed objects and retrieval of managed objects in both plaintext and optionally wrapped with another managed object. The KMIP protocol also includes support for a subset of the operations necessary for certificate management (certifying certificate requests and validating certificate changes). KMIP defines a range of Hash-based and MAC-based derivation options.

This document defines additional KMIP operations for cryptographic operations using managed objects for encryption, decryption, signature generation, signature verification, MAC generation, MAC verification, random number generation, and hashing. These operations are intended for use for key management operations.

KMIP clients and servers that support the cryptographic operations should be mindful of selecting the level of protection for the communication channel (the TLS connection) that provides sufficient protection of the plaintext data included in any of the cryptographic operations and commensurate with the security strength of the operation. There is no requirement for the KMIP server to enforce selection of a level of protection.

A KMIP server that supports the RNG Retrieve and RNG Seed operations may have a single RNG for the server, an RNG which is shared in an unspecified manner by KMIP clients or a separate RNG for each KMIP client. There is no requirement for the KMIP server to implement any specific RNG model.

The following sequence of operations outline the general use of cryptographic services by a KMIP client:

1. On each request
   a. Provide the Cryptographic Parameters (optional)
   b. Provide the Data (the input)
   c. Provide the IV/Counter/Nonce (optional).
   d. Use the Data from the response (the output)

Copyright © OASIS Open 2013. All Rights Reserved.

2.1 Encrypt Operation

[[Addition to [KMIP-SPEC-1_1] section 4 as a new Operation]]

This operation requests the server to perform an encryption operation on the provided data using a Managed Cryptographic Object as the key for the encryption operation.

The request contains information about the cryptographic parameters (mode and padding method), the data to be encrypted, and the IV/Counter/Nonce to use. The cryptographic parameters may be omitted from the request as they may be specified as associated attributes of the managed cryptographic object. The initialization vector/counter/nonce may also be omitted from the request if the cryptographic parameters indicate that the server shall generate a random IV on behalf of the client or the encryption algorithm does not require an IV/Counter/Nonce. The server does not store or otherwise manage the IV/Counter/Nonce.

If the Managed Cryptographic Object referenced has a Usage Limits attribute then the server SHALL obtain an allocation from the current Usage Limits value prior to performing the encryption operation. If the allocation is unable to be obtain the operation SHALL return with a result status of Operation Failed and result reason of Permission Denied.

The response contains the Unique Identifier of the Managed Cryptographic Object used as the key and the result of the encryption operation.

The success or failure of the operation is indicated by the Result Status (and if failure the Result Reason) in the response header.

Request Payload		
Object	**REQUIRED**	**Description**
Unique Identifier, see [KMIP-SPEC-1_1] 3.1	No	The Unique Identifier of the Managed Cryptographic Object that is the key to use for the encryption operation. If omitted, then the ID Placeholder value SHALL be used by the server as the Unique Identifier.
Cryptographic Parameters, see [KMIP-SPEC-1_1] 3.6	No	The Cryptographic Parameters (Block Cipher Mode, Padding Method, RandomIV) corresponding to the particular encryption method requested. If omitted then the Cryptographic Parameters associated with the Managed Cryptographic Object with the lowest Attribute Index SHALL be used. If there are no Cryptographic Parameters associated with the Managed Cryptographic Object and the algorithm requires parameters then the operation SHALL return with a Result Status of Operation Failed
Data	Yes	The data to be encrypted (as a Byte String).
IV/Counter/Nonce	No	The initialization vector, counter or nonce to be used (where appropriate).

Table 1: Encrypt Request Payload

Copyright © OASIS Open 2013. All Rights Reserved.

Response Payload		
Object	**REQUIRED**	**Description**
Unique Identifier, see [KMIP-SPEC-1_1] 3.1	Yes	The Unique Identifier of the Managed Cryptographic Object that was the key used for the encryption operation.
Data	Yes	The encrypted data (as a Byte String).
IV/Counter/Nonce	No	The value used if the Cryptographic Parameters specified Random IV and the IV/Counter/Nonce value was not provided in the request and the algorithm requires the provision of an IV/Counter/Nonce.

Table 2: Encrypt Response Payload

Copyright © OASIS Open 2013. All Rights Reserved.

2.2 Decrypt Operation

[[Addition to [KMIP-SPEC-1_1] section 4 as a new Operation]]

This operation requests the server to perform a decryption operation on the provided data using a Managed Cryptographic Object as the key for the decryption operation.

The request contains information about the cryptographic parameters (mode and padding method), the data to be decrypted, and the IV/Counter/Nonce to use. The cryptographic parameters may be omitted from the request as they may be specified as associated attributes of the managed cryptographic object. The initialization vector/counter/nonce may also be omitted from the request if the algorithm does not require an IV/Counter/Nonce.

The response contains the Unique Identifier of the Managed Cryptographic Object used as the key and the result of the decryption operation.

The success or failure of the operation is indicated by the Result Status (and if failure the Result Reason) in the response header.

Request Payload		
Object	**REQUIRED**	**Description**
Unique Identifier, see [KMIP-SPEC-1_1] 3.1	No	The Unique Identifier of the Managed Cryptographic Object that is the key to use for the decryption operation. If omitted, then the ID Placeholder value SHALL be used by the server as the Unique Identifier.
Cryptographic Parameters, see [KMIP-SPEC-1_1] 3.6	No	The Cryptographic Parameters (Block Cipher Mode, Padding Method) corresponding to the particular decryption method requested. If omitted then the Cryptographic Parameters associated with the Managed Cryptographic Object with the lowest Attribute Index SHALL be used. If there are no Cryptographic Parameters associated with the Managed Cryptographic Object and the algorithm requires parameters then the operation SHALL return with a Result Status of Operation Failed
Data	Yes	The data to be decrypted (as a Byte String).
IV/Counter/Nonce	No	The initialization vector, counter or nonce to be used (where appropriate).

Table 3: Decrypt Request Payload

Response Payload		
Object	**REQUIRED**	**Description**
Unique Identifier, see [KMIP-SPEC-1_1] 3.1	Yes	The Unique Identifier of the Managed Cryptographic Object that is the key used for the decryption operation.
Data	Yes	The decrypted data (as a Byte String).

Table 4: Decrypt Response Payload

Copyright © OASIS Open 2013. All Rights Reserved.

2.3 Sign Operation

[[Addition to [KMIP-SPEC-1_1] section 4 as a new Operation]]

This operation requests the server to perform a signature operation on the provided data using a Managed Cryptographic Object as the key for the signature operation.

The request contains information about the cryptographic parameters (digital signature algorithm or cryptographic algorithm and hash algorithm) and the data to be signed. The cryptographic parameters may be omitted from the request as they may be specified as associated attributes of the managed cryptographic object.

If the Managed Cryptographic Object referenced has a Usage Limits attribute then the server SHALL obtain an allocation from the current Usage Limits value prior to performing the signing operation. If the allocation is unable to be obtain the operation SHALL return with a result status of Operation Failed and result reason of Permission Denied.

The response contains the Unique Identifier of the Managed Cryptographic Object used as the key and the result of the signature operation.

The success or failure of the operation is indicated by the Result Status (and if failure the Result Reason) in the response header.

Request Payload		
Object	**REQUIRED**	**Description**
Unique Identifier, see [KMIP-SPEC-1_1] 3.1	No	The Unique Identifier of the Managed Cryptographic Object that is the key to use for the signature operation. If omitted, then the ID Placeholder value SHALL be used by the server as the Unique Identifier.
Cryptographic Parameters, see [KMIP-SPEC-1_1] 3.6	No	The Cryptographic Parameters (Digital Signature Algorithm or Cryptographic Algorithm and Hashing Algorithm) corresponding to the particular signature generation method requested. If omitted then the Cryptographic Parameters associated with the Managed Cryptographic Object with the lowest Attribute Index SHALL be used. If there are no Cryptographic Parameters associated with the Managed Cryptographic Object and the algorithm requires parameters then the operation SHALL return with a Result Status of Operation Failed
Data	Yes	The data to be signed (as a Byte String).

Table 5: Sign Request Payload

Copyright © OASIS Open 2013. All Rights Reserved.

Response Payload		
Object	**REQUIRED**	**Description**
Unique Identifier, see [KMIP-SPEC-1_1] 3.1	Yes	The Unique Identifier of the Managed Cryptographic Object that is the key used for the signature operation.
Data	Yes	The signed data (as a Byte String).

Table 6: Sign Response Payload

Copyright © OASIS Open 2013. All Rights Reserved.

2.4 Signature Verify Operation

[[Addition to [KMIP-SPEC-1_1] section 4 as a new Operation]]

This operation requests the server to perform a signature verify operation on the provided data using a Managed Cryptographic Object as the key for the signature verification operation.

The request contains information about the cryptographic parameters (digital signature algorithm or cryptographic algorithm and hash algorithm) and the signature to be verified and optionally the data that was passed to the signing operation (for those algorithms which require the original data to verify a signature).

The cryptographic parameters may be omitted from the request as they may be specified as associated attributes of the managed cryptographic object.

The response contains the Unique Identifier of the Managed Cryptographic Object used as the key and the optional data recovered from the signature (for those signature algorithms where data recovery from the signature is supported). The validity of the signature is indicated by the Validity Indicator field.

The success or failure of the operation is indicated by the Result Status (and if failure the Result Reason) in the response header.

Request Payload		
Object	**REQUIRED**	**Description**
Unique Identifier, see [KMIP-SPEC-1_1] 3.1	No	The Unique Identifier of the Managed Cryptographic Object that is the key to use for the signature verify operation. If omitted, then the ID Placeholder value SHALL be used by the server as the Unique Identifier.
Cryptographic Parameters, see [KMIP-SPEC-1_1] 3.6	No	The Cryptographic Parameters (Digital Signature Algorithm or Cryptographic Algorithm and Hashing Algorithm) corresponding to the particular signature verification method requested. If omitted then the Cryptographic Parameters associated with the Managed Cryptographic Object with the lowest Attribute Index SHALL be used. If there are no Cryptographic Parameters associated with the Managed Cryptographic Object and the algorithm requires parameters then the operation SHALL return with a Result Status of Operation Failed
Data	No	The data that was signed (as a Byte String).
Signature Data	Yes	The signature to be verified (as a Byte String).

Table 7: Signature Verify Request Payload

Copyright © OASIS Open 2013. All Rights Reserved.

Response Payload		
Object	**REQUIRED**	**Description**
Unique Identifier, see [KMIP-SPEC-1_1] 3.1	Yes	The Unique Identifier of the Managed Cryptographic Object that is the key used for the verification operation.
Validity Indicator, see [KMIP-SPEC-1_1] 9.1.3.2.23	Yes	An Enumeration object indicating whether the signature is valid, invalid, or unknown.
Data	No	The optional recovered data (as a Byte String) for those signature algorithms where data recovery from the signature is supported.

Table 8: Signature Verify Response Payload

Copyright © OASIS Open 2013. All Rights Reserved.

2.5 MAC Operation

[[Addition to [KMIP-SPEC-1_1] section 4 as a new Operation]]

This operation requests the server to perform message authentication code (MAC) operation on the provided data using a Managed Cryptographic Object as the key for the MAC operation.

The request contains information about the cryptographic parameters (cryptographic algorithm) and the data to be MACed. The cryptographic parameters may be omitted from the request as they may be specified as associated attributes of the managed cryptographic object.

The response contains the Unique Identifier of the Managed Cryptographic Object used as the key and the result of the MAC operation.

The success or failure of the operation is indicated by the Result Status (and if failure the Result Reason) in the response header.

Request Payload		
Object	**REQUIRED**	**Description**
Unique Identifier, see [KMIP-SPEC-1_1] 3.1	No	The Unique Identifier of the Managed Cryptographic Object that is the key to use for the MAC operation. If omitted, then the ID Placeholder value SHALL be used by the server as the Unique Identifier.
Cryptographic Parameters, see [KMIP-SPEC-1_1] 3.6	No	The Cryptographic Parameters (Cryptographic Algorithm) corresponding to the particular MAC method requested. If omitted then the Cryptographic Parameters associated with the Managed Cryptographic Object with the lowest Attribute Index SHALL be used. If there are no Cryptographic Parameters associated with the Managed Cryptographic Object and the algorithm requires parameters then the operation SHALL return with a Result Status of Operation Failed
Data	Yes	The data to be MACed (as a Byte String).

Table 9: MAC Request Payload

Response Payload		
Object	**REQUIRED**	**Description**
Unique Identifier, see [KMIP-SPEC-1_1] 3.1	Yes	The Unique Identifier of the Managed Cryptographic Object that is the key used for the MAC operation.
Data	Yes	The data MACed (as a Byte String).

Table 10: MAC Response Payload

Copyright © OASIS Open 2013. All Rights Reserved.

2.6 MAC Verify Operation

[[Addition to [KMIP-SPEC-1_1] section 4 as a new Operation]]

This operation requests the server to perform message authentication code (MAC) verify operation on the provided data using a Managed Cryptographic Object as the key for the MAC verify operation.

The request contains information about the cryptographic parameters (cryptographic algorithm) and the data to be MAC verified. The cryptographic parameters may be omitted from the request as they may be specified as associated attributes of the managed cryptographic object.

The response contains the Unique Identifier of the Managed Cryptographic Object used as the key and the result of the MAC verify operation. The validity of the MAC is indicated by the Validity Indicator field.

The success or failure of the operation is indicated by the Result Status (and if failure the Result Reason) in the response header.

Request Payload		
Object	**REQUIRED**	**Description**
Unique Identifier, see [KMIP-SPEC-1_1] 3.1	No	The Unique Identifier of the Managed Cryptographic Object that is the key to use for the MAC verify operation. If omitted, then the ID Placeholder value SHALL be used by the server as the Unique Identifier.
Cryptographic Parameters, see [KMIP-SPEC-1_1] 3.6	No	The Cryptographic Parameters (Cryptographic Algorithm) corresponding to the particular MAC method requested. If omitted then the Cryptographic Parameters associated with the Managed Cryptographic Object with the lowest Attribute Index SHALL be used. If there are no Cryptographic Parameters associated with the Managed Cryptographic Object and the algorithm requires parameters then the operation SHALL return with a Result Status of Operation Failed
Data	Yes	The data to be MAC verified (as a Byte String).

Table 11: MAC Verify Request Payload

Response Payload		
Object	**REQUIRED**	**Description**
Unique Identifier, see [KMIP-SPEC-1_1] 3.1	Yes	The Unique Identifier of the Managed Cryptographic Object that is the key used for the verification operation.
Validity Indicator, see [KMIP-SPEC-1_1] 9.1.3.2.23	Yes	An Enumeration object indicating whether the MAC is valid, invalid, or unknown.

Table 12: MAC Verify Response Payload

Copyright © OASIS Open 2013. All Rights Reserved.

2.7 RNG Retrieve Operation

[[Addition to [KMIP-SPEC-1_1] section 4 as a new Operation]]

This operation requests the server to return output from a Random Number Generator (RNG).

The request contains the quantity of output required.

The response contains the RNG output.

The success or failure of the operation is indicated by the Result Status (and if failure the Result Reason) in the response header.

Request Payload		
Object	**REQUIRED**	**Description**
Data Length	Yes	The amount of random number generator output to be returned (in bytes)

Table 13: RNG Retrieve Request Payload

Response Payload		
Object	**REQUIRED**	**Description**
Data	Yes	The random number generator output

Table 14: RNG Retrieve Response Payload

Copyright © OASIS Open 2013. All Rights Reserved.

2.8 RNG Seed Operation

[[Addition to [KMIP-SPEC-1_1] section 4 as a new Operation]]

This operation requests the server to seed a Random Number Generator.

The request contains the seeding material.

The response optionally contains the amount of seed data used. ~~RNG algorithm (which may not have been provided in the request)~~.

The success or failure of the operation is indicated by the Result Status (and if failure the Result Reason) in the response header.

The server may elect to ignore the information provided by the client (i.e. not accept the seeding material) and MAY indicate this to the client by returning zero as the value in the Data Length response. A client SHALL NOT consider a response from a server which does not use the provided data as an error.

Request Payload		
Object	**REQUIRED**	**Description**
Data	Yes	The data to be provided as a seed to the random number generator

Table 15: RNG Seed Request Payload

Response Payload		
Object	**REQUIRED**	**Description**
Data Length	Yes	The amount of seed data used (in bytes)

Table 16: RNG Seed Response Payload

Copyright © OASIS Open 2013. All Rights Reserved.

2.9 HASH Operation

[[Addition to [KMIP-SPEC-1_1] section 4 as a new Operation]]

This operation requests the server to perform a hash operation on the data provided-.

The request contains information about the cryptographic parameters (hash algorithm) and the data to be hashed.

The response contains the result of the hash operation.

The success or failure of the operation is indicated by the Result Status (and if failure the Result Reason) in the response header.

Request Payload		
Object	**REQUIRED**	**Description**
Cryptographic Parameters, see [KMIP-SPEC-1_1] 3.6	Yes	The Cryptographic Parameters (Hashing Algorithm) corresponding to the particular hash method requested.
Data	Yes	The data to be hashed (as a Byte String).

Table 17: MAC Request Payload

Response Payload		
Object	**REQUIRED**	**Description**
Data	Yes	The hashed data (as a Byte String).

Table 18: HASH Response Payload

Copyright © OASIS Open 2013. All Rights Reserved.

2.10 Message Encoding

The following additions are required to [KMIP-SPEC-1_1] to define the tag values and operation enumeration values required for this profile.

2.10.1 Cryptographic Parameters

[[Updates to [KMIP-SPEC-1_1] 3.6 adding paragraphs and updating the table with three new rows at the end]]

The Cryptographic Algorithm is also used to specify the parameters for the cryptographic operations defined in the Cryptographic Profile. For operations involving digital signatures either the Digital Signature Algorithm can be specified or the Cryptographic Algorithm and Hashing Algorithm combination.

Random IV can be used to request that the KMIP server generate an appropriate IV for a cryptographic operation that requires an IV. The generated Random IV is returned in the response to the cryptographic operation.

Object	Encoding	REQUIRED
Cryptographic Parameters	Structure	
Block Cipher Mode	Enumeration, see 9.1.3.2.14	No
Padding Method	Enumeration, see 9.1.3.2.15	No
Hashing Algorithm	Enumeration, see 9.1.3.2.16	No
Key Role Type	Enumeration, see 9.1.3.2.17	No
Digital Signature Algorithm	Enumeration, see 9.1.3.2.7	No
Cryptographic Algorithm	Enumeration, see 9.1.3.2.13	No
Random IV	Boolean	No

2.10.2 Data

[[Addition to [KMIP-SPEC-1_1] section 2.1 as a new Base Object]]

This is used in requests and responses in cryptographic operations that require data to be passed between the client and the server.

Object	Encoding
Data	Byte String

2.10.3 Data Length

[[Addition to [KMIP-SPEC-1_1] section 2.1 as a new Base Object]]

This is used in requests in cryptographic operations to indicate the amount of data expected in a response.

Copyright © OASIS Open 2013. All Rights Reserved.

Object	Encoding
Data Length	Integer

2.10.4 Signature Data

[[Addition to [KMIP-SPEC-1_1] section 2.1 as a new Base Object]]

This is used in requests and responses in cryptographic operations that require signature data to be passed between the client and the server.

Object	Encoding
Signature Data	Byte String

2.10.5 Tags

Addition to [KMIP-SPEC-1_1] 9.1.3.1

Tag	
Object	**Tag Value**
Data	4200B8
Signature Data	4200B9
Data Length	4200BA
Random IV	4200BB

2.10.6 Operation Enumeration

Addition to [KMIP-SPEC-1_1] 9.1.3.2.27

Operation	
Name	**Value**
Encrypt	0000001F
Decrypt	00000020
Sign	00000021
Signature Verify	00000022
MAC	00000023
MAC Verify	00000024
RNG Retrieve	00000025
RNG Seed	00000026
HASH	00000027

Copyright © OASIS Open 2013. All Rights Reserved.

2.10.7 Conformance Clauses

Implementations conformant to this profile SHALL support one or more of the base profiles defined within section 3 of [KMIP-PROF] along with one or more conformance clauses including the sub-clauses of each clause below.

2.11 Base Crypto Server Clause

This proposal builds on the KMIP server conformance clauses in [KMIP-PROF] to provide the most basic functionality that would be expected of a conformant KMIP server – the ability to accept requests for encryption and decryption operations from a KMIP client.

2.11.1 Implementation Conformance

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

2.11.2 Conformance of a Base Crypto Server

An implementation conforms to this specification as a Base Crypto 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 Encrypt client-to-server operation (2.1)
3. Supports the Decrypt client-to-server operation (2.2)

2.12 Base Crypto Client Clause

This proposal builds on the KMIP client conformance clauses in [KMIP-PROF] to provide the most basic functionality that would be expected of a conformant KMIP client – the ability to request encryption and decryption operations from a KMIP server.

2.12.1 Implementation Conformance

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

2.12.2 Conformance of a Base Crypto Client

An implementation conforms to this specification as a Base Crypto 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 Encrypt and/or Decrypt client-to-server operation (2.1 and/or 2.2)

Copyright © OASIS Open 2013. All Rights Reserved.

2.13 RNG Crypto Server Clause

This proposal builds on the KMIP server conformance clauses in [KMIP-PROF] to provide the most basic functionality that would be expected of a conformant KMIP server – the ability to accept requests for RNG operations from a KMIP client.

2.13.1 Implementation Conformance

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

2.13.2 Conformance of a RNG Crypto Server

An implementation conforms to this specification as a RNG Crypto 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 RNG Retrieve client-to-server operation (2.7)
3. Supports the RNG Seed client-to-server operation (2.8)

2.14 RNG Crypto Client Clause

This proposal builds on the KMIP client conformance clauses in [KMIP-PROF] to provide the most basic functionality that would be expected of a conformant KMIP RNG client – the ability to request RNG output from a KMIP server.

2.14.1 Implementation Conformance

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

2.14.2 Conformance of a RNG Crypto Client

An implementation conforms to this specification as a Base Crypto 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 RNG Retrieve and/or RNG Seed client-to-server operation (2.7 and/or 2.8)

Copyright © OASIS Open 2013. All Rights Reserved.

2.15 Advanced Crypto Server Clause

This proposal builds on the KMIP server conformance clauses in [KMIP-PROF] to provide advanced functionality that would be expected of a conformant KMIP client – the ability to request encryption, decryption, signature, and verification operations from a KMIP client.

2.15.1 Implementation Conformance

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

2.15.2 Conformance of an Advanced Crypto Server

An implementation conforms to this specification as an Advanced Crypto 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 Encrypt client-to-server operation (2.1)
3. Supports the Decrypt client-to-server operation (2.2)
4. Supports the Sign client-to-server operation (2.3)
5. Supports the Signature Verify client-to-server operation (2.4)
6. Supports the MAC client-to-server operation (2.5)
7. Supports the MAC Verify client-to-server operation (2.6)
8. Supports the RNG Retrieve client-to-server operation (2.7)
9. Supports the RNG Seed client-to-server operation (2.8)

2.16 Advanced Crypto Client Clause

This proposal builds on the KMIP client conformance clauses in [KMIP-PROF] to provide the advanced functionality that would be expected of a conformant KMIP client – the ability to request encryption, decryption, signature, and verification operations from a KMIP server.

2.16.1 Implementation Conformance

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

2.16.2 Conformance of an Advanced Crypto Client

An implementation conforms to this specification as an Advanced Crypto 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 Encrypt and/or Decrypt client-to-server operation (2.1 and/or 2.2)
3. Supports the Sign and/or Signature Verify client-to-server operation (2.3 and/or 2.4)
4. Supports the MAC and/or MAC Verify client-to-server operation (2.5 and/or 2.6)
5. Supports the RNG Retrieve and/or RNG Seed client-to-server operation (2.7 and/or 2.8)

Copyright © OASIS Open 2013. All Rights Reserved.

3 Crypto Profile Test Cases

This section contains a test case that demonstrates the noted Crypto Profile.

Note: the specifics of the request and response messages for each test have not yet been documented.

3.1 Base Crypto Tests

A KMIP client supporting the Base Crypto Profile supports one or more of the following tests.

A KMIP server supporting the Base Crypto Profile supports all of the following tests.

3.1.1 Test Case: Encrypt with Known Symmetric Key

Register a symmetric key and perform encrypt using the symmetric key.

Time	Request/Response messages
0	Register
1	Encrypt
2	Destroy

3.1.2 Test Case: Decrypt with Known Symmetric Key

Register a symmetric key and perform decrypt using the symmetric key.

Time	Request/Response messages
0	Register
1	Decrypt
2	Destroy

3.1.3 Test Case: Encrypt and Decrypt with Known Symmetric Key

Register a symmetric key and perform both encrypt and decrypt operations using the symmetric key.

Time	Request/Response messages
0	Register
1	Encrypt
2	Decrypt
3	Destroy

3.1.4 Test Case: Encrypt with New Symmetric Key

Create a symmetric key and perform encrypt using the symmetric key.

Time	Request/Response messages
0	Create
1	Encrypt
2	Destroy

Copyright © OASIS Open 2013. All Rights Reserved.

3.1.5 Test Case: Decrypt with New Symmetric Key

Create a symmetric key and perform decrypt using the symmetric key.

Note: Create followed by Decrypt is unusual but some applications actually do this relying on Decrypt and Encrypt being able to be used around the wrong way to get the same result.

Time	Request/Response messages
0	Create
1	Decrypt
2	Destroy

3.1.6 Test Case: Encrypt and Decrypt with New Symmetric Key

Create a symmetric key and perform both encrypt and decrypt operations using the symmetric key.

Time	Request/Response messages
0	Create
1	Encrypt
2	Decrypt
3	Destroy

3.2 RNG Crypto Tests

A KMIP client supporting the RNG Crypto Profile supports one or more of the following tests.

A KMIP server supporting the RNG Crypto Profile supports all of the following tests.

3.2.1 Test Case: RNG Seed

Seed an RNG.

Time	Request/Response messages
0	RNG Seed

3.2.2 Test Case: RNG Retrieve

Retrieve output from an RNG.

Time	Request/Response messages
0	RNG Retrieve

Copyright © OASIS Open 2013. All Rights Reserved.

3.3 Advanced Crypto Tests

A KMIP client supporting the Advanced Crypto Profile supports one or more of the following tests in addition to one or more of the RNG Crypto Tests and one or more of the Base Crypto Tests.

A KMIP server supporting the Advanced Crypto Profile supports all of the following tests in addition to all of the RNG Crypto Tests and all of the Base Crypto Tests.

3.3.1 Test Case: Sign with Known Asymmetric Key

Register an asymmetric key and perform sign using the asymmetric key.

Time	Request/Response messages
0	Register
1	Sign
2	Destroy

3.3.2 Test Case: Signature Verify with Known Asymmetric Key

Register an asymmetric key and perform signature verification using the asymmetric key.

Time	Request/Response messages
0	Register
1	Signature Verify
2	Destroy

3.3.3 Test Case: Sign and Signature Verify with Known Asymmetric Key

Register an asymmetric key and perform both and signature verification operations using the asymmetric key.

Time	Request/Response messages
0	Register
1	Sign
2	Signature Verify
3	Destroy

3.3.4 MAC with Known Key

Register a key and perform MAC operations using the key.

Time	Request/Response messages
0	Register
1	MAC
2	Destroy

Copyright © OASIS Open 2013. All Rights Reserved.

3.3.5 MAC Verify with Known Key

Register a key and perform MAC verification operations using the key.

Time	Request/Response messages
0	`Register`
1	`MAC`
2	`Destroy`

3.3.6 Test Case: MAC and MAC Verify with Known Key

Register a key and perform both MAC and MAC verify operations using the key.

Time	Request/Response messages
0	`Register`
1	`MAC`
2	`MAC Verify`
3	`Destroy`

3.3.7 Test Case: HASH

Hash Data.

Time	Request/Response messages
0	`Hash`

Copyright © OASIS Open 2013. All Rights Reserved.

Appendix A. Acknowledgments

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

Elaine Barker, NIST
Kelley Burgin, NSA
Tony Cox, Cryptsoft
Indra Fitzgerald, Hewlett-Packard
Robert Griffin, EMC Corporation
John Leiseboer, Quintessence Labs
Bob Lockhart, Thales e-Security
Denis Pochuev, SafeNet
Subhash Sankuratripati, NetApp
Kiran Kumar Thota, VMware

Copyright © OASIS Open 2013. All Rights Reserved.