1.2 Normative References

[Bjoerkqvist2010] Mathias Björkqvist, Christian Cachin, Robert Haas, Xiao-Yu Hu, Anil Kurmus, René Pawlitzek, and Marko Vukolić. Design and implementation of a key-lifecycle management system. In Radu Sion, editor, *Proc. Financial Cryptography and Data Security (FC 2010)*, volume 6052 of *Lecture Notes in Computer Science*, pp160-174, 2010

[Cachin2009] C. Cachin *and N. Chandran.* A secure cryptographic token interface. In *Proc. Computer Security Foundations Symposium (CSF-22)*, pp 141-153. July 2009.

[FIPS140-2] *Security Requirements for Cryptographic Modules*, FIPS PUB 140-2, May 2001, http://csrc.nist.gov/publications/fips/fips140-2/fips1402.pdf

[FIPS186-3] *Digital Signature Standard (DSS)*, FIPS PUB 186-3, Jun 2009, http://csrc.nist.gov/publications/fips/fips186-3/fips_186-3.pdf

[FIPS197] *Advanced Encryption Standard*, FIPS PUB 197, Nov 2001, http://csrc.nist.gov/publications/fips/fips197/fips-197.pdf

[FIPS198-1] *The Keyed-Hash Message Authentication Code (HMAC)*, FIPS PUB 198-1, Jul 2008, http://csrc.nist.gov/publications/fips/fips198-1/FIPS-198-1_final.pdf

[IEEE1003-1] IEEE Std 1003.1, *Standard for information technology - portable operating system interface (POSIX). Shell and utilities*, 2004.

[ISO16609] ISO, *Banking -- Requirements for message authentication using symmetric techniques*, ISO 16609, 1991

[ISO9797-1] ISO/IEC, *Information technology -- Security techniques -- Message Authentication Codes (MACs) -- Part 1: Mechanisms using a block cipher*, ISO/IEC 9797-1, 1999

[KMIP-Prof] OASIS Standard, Key Management Interoperability Protocol Profiles Version 1.0, October 2010, http://docs.oasis-open.org/kmip/profiles/v1.0/os/kmip-profiles-1.0-os.doc

[PKCS#1] RSA Laboratories, *PKCS #1 v2.1: RSA Cryptography Standard*, Jun 14, 2002, http://www.rsa.com/rsalabs/node.asp?id=2125

[PKCS#5] RSA Laboratories, *PKCS #5 v2.1: Password-Based Cryptography Standard*, Oct 5, 2006, http://www.rsa.com/rsalabs/node.asp?id=2127

[PKCS#7] RSA Laboratories, *PKCS#7 v1.5: Cryptographic Message Syntax Standard,* Nov 1, 1993, http://www.rsa.com/rsalabs/node.asp?id=2129

[PKCS#8] RSA Laboratories, *PKCS#8 v1.2: Private-Key Information Syntax Standard*, Nov 1, 1993, http://www.rsa.com/rsalabs/node.asp?id=2130

[PKCS#10] RSA Laboratories, *PKCS #10 v1.7: Certification Request Syntax Standard*, May 26, 2000, http://www.rsa.com/rsalabs/node.asp?id=2132

[RFC1319] B. Kaliski, *The MD2 Message-Digest Algorithm*, IETF RFC 1319, Apr 1992, http://www.ietf.org/rfc/rfc1319.txt

[RFC1320] R. Rivest, *The MD4 Message-Digest Algorithm*, IETF RFC 1320, Apr 1992, http://www.ietf.org/rfc/rfc1320.txt

[RFC1321] R. Rivest, *The MD5 Message-Digest Algorithm*, IETF RFC 1321, Apr 1992, http://www.ietf.org/rfc/rfc1321.txt

[RFC1421] J. Linn, *Privacy Enhancement for Internet Electronic Mail: Part I: Message Encryption and Authentication Procedures*, IETF RFC 1421, Feb 1993, http://www.ietf.org/rfc/rfc1421.txt

[RFC1424] B. Kaliski, *Privacy Enhancement for Internet Electronic Mail: Part IV: Key Certification and Related Services*, IETF RFC 1424, Feb 1993, http://www.ietf.org/rfc/rfc1424.txt

[RFC2104] H. Krawczyk, M. Bellare, R. Canetti, *HMAC: Keyed-Hashing for Message Authentication*, IETF RFC 2104, Feb 1997, http://www.ietf.org/rfc/rfc2104.txt

[RFC2119] S. Bradner, *Key words for use in RFCs to Indicate Requirement Levels*, IETF RFC 2119, Mar 1997, http://www.ietf.org/rfc/rfc2119.txt

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

[RFC2898] B. Kaliski, *PKCS #5: Password-Based Cryptography Specification Version 2.0*, IETF RFC 2898, Sep 2000, http://www.ietf.org/rfc/rfc2898.txt

[RFC 3394] J. Schaad, R. Housley, *Advanced Encryption Standard (AES) Key Wrap Algorithm*, IETF RFC 3394, Sep 2002, http://www.ietf.org/rfc/rfc3394.txt

[RFC3447] J. Jonsson, B. Kaliski, *Public-Key Cryptography Standards (PKCS) #1: RSA Cryptography Specifications Version 2.1*, IETF RFC 3447, Feb 2003, http://www.ietf.org/rfc/rfc3447.txt

[RFC3629] F. Yergeau, *UTF-8, a transformation format of ISO 10646*, IETF RFC 3629, Nov 2003, http://www.ietf.org/rfc/rfc3629.txt

[RFC3647] S. Chokhani, W. Ford, R. Sabett, C. Merrill, and S. Wu, *Internet X.509 Public Key Infrastructure Certificate Policy and Certification Practices Framework*, IETF RFC 3647, Nov 2003, http://www.ietf.org/rfc/rfc3647.txt

[RFC4210] C. Adams, S. Farrell, T. Kause and T. Mononen, *Internet X.509 Public Key Infrastructure Certificate Management Protocol (CMP)*, IETF RFC 2510, Sep 2005, http://www.ietf.org/rfc/rfc4210.txt

[RFC4211] J. Schaad, *Internet X.509 Public Key Infrastructure Certificate Request Message Format (CRMF),* IETF RFC 4211, Sep 2005, http://www.ietf.org/rfc/rfc4211.txt

[RFC4868] S. Kelly, S. Frankel, *Using HMAC-SHA-256, HMAC-SHA-384, and HMAC-SHA-512 with IPsec*, IETF RFC 4868, May 2007, http://www.ietf.org/rfc/rfc4868.txt

[RFC4880] J. Callas, L. Donnerhacke, H. Finney, D. Shaw, and R. Thayer, *OpenPGP Message Format*, IETF RFC 4880, Nov 2007, http://www.ietf.org/rfc/rfc4880.txt

[RFC4949] R. Shirey, *Internet Security Glossary, Version 2*, IETF RFC 4949, Aug 2007, http://www.ietf.org/rfc/rfc4949.txt

[RFC5272] J. Schaad and M. Meyers, *Certificate Management over CMS (CMC)*, IETF RFC 5272, Jun 2008, http://www.ietf.org/rfc/rfc5272.txt

[RFC5280] D. Cooper, S. Santesson, S. Farrell, S. Boeyen, R. Housley, W. Polk, *Internet X.509 Public Key Infrastructure Certificate*, IETF RFC 5280, May 2008, http://www.ietf.org/rfc/rfc5280.txt

[RFC5649] R. Housley, *Advanced Encryption Standard (AES) Key Wrap with Padding Algorithm*, IETF RFC 5649, Aug 2009, http://www.ietf.org/rfc/rfc5649.txt

[SHAMIR1979] A. Shamir, *How to share a secret*, Communications of the ACM, vol. 22, no. 11, pp. 612-613, Nov 1979

[SP800-38A] M. Dworkin, *Recommendation for Block Cipher Modes of Operation – Methods and Techniques*, NIST Special Publication 800-38A, Dec 2001, http://csrc.nist.gov/publications/nistpubs/800-38a/sp800-38a.pdf

[SP800-38B] M. Dworkin, *Recommendation for Block Cipher Modes of Operation: The CMAC Mode for Authentication*, NIST Special Publication 800-38B, May 2005, http://csrc.nist.gov/publications/nistpubs/800-38B/SP_800-38B.pdf

[SP800-38C] M. Dworkin, *Recommendation for Block Cipher Modes of Operation: the CCM Mode for Authentication and Confidentiality*, NIST Special Publication 800-38C, May 2004, http://csrc.nist.gov/publications/nistpubs/800-38C/SP800-38C_updated-July20_2007.pdf

[SP800-38D] M. Dworkin, *Recommendation for Block Cipher Modes of Operation: Galois/Counter Mode (GCM) and GMAC*, NIST Special Publication 800-38D, Nov 2007, http://csrc.nist.gov/publications/nistpubs/800-38D/SP-800-38D.pdf

[SP800-38E] M. Dworkin, *Recommendation for Block Cipher Modes of Operation: The XTS-AES Mode for Confidentiality on Block-Oriented Storage Devices*, NIST Special Publication 800-38E, Jan 2010, http://csrc.nist.gov/publications/nistpubs/800-38E/nist-sp-800-38E.pdf

[SP800-56A] E. Barker, D. Johnson, and M. Smid, *Recommendation for Pair-Wise Key Establishment Schemes Using Discrete Logarithm Cryptography (Revised)*, NIST Special Publication 800-56A, Mar 2007, http://csrc.nist.gov/publications/nistpubs/800-56A/SP800-56A_Revision1_Mar08-2007.pdf

[SP800-56B] E. Barker, L. Chen, A. Regenscheid, and M. Smid, *Recommendation for Pair-Wise Key Establishment Schemes Using Integer Factorization Cryptography*, NIST Special Publication 800-56B, Aug 2009, http://csrc.nist.gov/publications/nistpubs/800-56B/sp800-56B.pdf

[SP800-57-1] E. Barker, W. Barker, W. Burr, W. Polk, and M. Smid, *Recommendations for Key Management - Part 1: General (Revised),* NIST Special Publication 800-57 part 1, Mar 2007, http://csrc.nist.gov/publications/nistpubs/800-57/sp800-57-Part1-revised2_Mar08-2007.pdf

[SP800-67] W. Barker, *Recommendation for the Triple Data Encryption Algorithm (TDEA) Block Cipher*, NIST Special Publication 800-67, Version 1.1, Revised 19 May 2008, http://csrc.nist.gov/publications/nistpubs/800-67/SP800-67.pdf

[SP800-108] L. Chen, *Recommendation for Key Derivation Using Pseudorandom Functions (Revised)*, NIST Special Publication 800-108, Oct 2009, http://csrc.nist.gov/publications/nistpubs/800-108/sp800-108.pdf

[X.509] International Telecommunication Union (ITU)–T, *X.509: Information technology – Open systems interconnection – The Directory: Public-key and attribute certificate frameworks*, Aug 2005, http://www.itu.int/rec/T-REC-X.509-200508-I/en

[X9.24-1] ANSI, *X9.24 - Retail Financial Services Symmetric Key Management - Part 1: Using Symmetric Techniques*, 2004.

[X9.31] ANSI, *X9.31:Digital Signatures Using Reversible Public Key Cryptography for the Financial Services Industry (rDSA)*, Sep 1998.

[X9.42] ANSI, *X9-42: Public Key Cryptography for the Financial Services Industry: Agreement of Symmetric Keys Using Discrete Logarithm Cryptography*, 2003.

[X9-57] ANSI, *X9-57: Public Key Cryptography for the Financial Services Industry: Certificate Management*, 1997.

[X9.62] ANSI, *X9-62: Public Key Cryptography for the Financial Services Industry, The Elliptic Curve Digital Signature Algorithm (ECDSA)*, 2005.

[X9-63] ANSI, *X9-63: Public Key Cryptography for the Financial Services Industry, Key Agreement and Key Transport Using Elliptic Curve Cryptography*, 2001.

[X9-102] ANSI, *X9-102: Symmetric Key Cryptography for the Financial Services Industry - Wrapping of Keys and Associated Data*, 2008.

[X9 TR-31] ANSI, *X9 TR-31: Interoperable Secure Key Exchange Key Block Specification for Symmetric Algorithms*, 2005.

3.13 Operation Policy Name

An operation policy controls what entities MAY perform which key management operations on the object. The content of the *Operation Policy Name* attribute is the name of a policy object known to the key management system and, therefore, is server dependent. The named policy objects are created and managed using mechanisms outside the scope of the protocol. The policies determine what entities MAY perform specified operations on the object, and which of the object's attributes MAY be modified or deleted. The Operation Policy Name attribute SHOULD be set when operations that result in a new Managed Object on the server are executed. It is set either explicitly or via some default set by the server, which then applies the named policy to all subsequent operations on the object.

Object	Encoding	
Operation Policy Name	Text String	

Table 60: Operation Policy Name Attribute

SHALL always have a value	No
Initially set by	Server or Client
Modifiable by server	Yes
Modifiable by client	No
Deletable by client	No
Multiple instances permitted	No
When implicitly set	Create, Create Key Pair, Register, Derive Key, Certify, Re-certify, Re-key
Applies to Object Types	All Objects

Table 61: Operation Policy Name Attribute Rules

3.13.1 Operations outside of operation policy control

Some of the operations SHOULD be allowed for any client at any time, without respect to operation policy. These operations are:

- Create
- Create Key Pair
- Register
- Certify
- Re-certify
- Validate
- Query
- Cancel
- Poll

3.13.2 Default Operation Policy

A key management system implementation SHALL implement at least one named operation policy, which is used for objects when the *Operation Policy* attribute is not specified by the Client in operations that result in a new Managed Object on the server, or in a template specified in these operations. This policy

is named *default*. It specifies the following rules for operations on objects created or registered with this policy, depending on the object type. ~~For the profiles defined in [KMIP-Prof], the creator SHALL be as defined in [KMIP-Prof].~~

3.13.2.1 Default Operation Policy for Secret Objects

This policy applies to Symmetric Keys, Private Keys, Split Keys, Secret Data, and Opaque Objects.

Default Operation Policy for Secret Objects	
Operation	**Policy**
Re-Key	Allowed to Owner~~creator~~ only
Derive Key	Allowed to Owner~~creator~~ only
Locate	Allowed to Owner~~creator~~ only
Check	Allowed to Owner~~creator~~ only
Get	Allowed to Owner~~creator~~ only
Get Attributes	Allowed to Owner~~creator~~ only
Get Attribute List	Allowed to Owner~~creator~~ only
Add Attribute	Allowed to Owner~~creator~~ only
Modify Attribute	Allowed to Owner~~creator~~ only
Delete Attribute	Allowed to Owner~~creator~~ only
Obtain Lease	Allowed to Owner~~creator~~ only
Get Usage Allocation	Allowed to Owner~~creator~~ only
Activate	Allowed to Owner~~creator~~ only
Revoke	Allowed to Owner~~creator~~ only
Destroy	Allowed to Owner~~creator~~ only
Archive	Allowed to Owner~~creator~~ only
Recover	Allowed to Owner~~creator~~ only

Table 62: Default Operation Policy for Secret Objects

3.13.2.2 Default Operation Policy for Certificates and Public Key Objects

This policy applies to Certificates and Public Keys.

Default Operation Policy for Certificates and Public Key Objects	
Operation	**Policy**
Locate	Allowed to all
Check	Allowed to all
Get	Allowed to all
Get Attributes	Allowed to all
Get Attribute List	Allowed to all
Add Attribute	Allowed to Ownercreator only
Modify Attribute	Allowed to Ownercreator only
Delete Attribute	Allowed to Ownercreator only
Obtain Lease	Allowed to all
Activate	Allowed to Ownercreator only
Revoke	Allowed to Ownercreator only
Destroy	Allowed to Ownercreator only
Archive	Allowed to Ownercreator only
Recover	Allowed to Ownercreator only

Table 63: Default Operation Policy for Certificates and Public Key Objects

3.13.2.3 Default Operation Policy for Template Objects

The operation policy specified as an attribute in the *Register* operation for a template object is the operation policy used for objects created using that template, and is not the policy used to control operations on the template itself. There is no mechanism to specify a policy used to control operations on template objects, so the default policy for template objects is always used for templates created by clients using the *Register* operation to create template objects.

Default Operation Policy for Private Template Objects	
Operation	**Policy**
Locate	Allowed to Ownercreator only
Get	Allowed to Ownercreator only
Get Attributes	Allowed to Ownercreator only
Get Attribute List	Allowed to Ownercreator only
Add Attribute	Allowed to Ownercreator only
Modify Attribute	Allowed to Ownercreator only
Delete Attribute	Allowed to Ownercreator only
Destroy	Allowed to Ownercreator only
Any operation referencing the Template using a Template-Attribute	Allowed to Ownercreator only

Table 64: Default Operation Policy for Private Template Objects

In addition to private template objects (which are controlled by the above policy, and which MAY be created by clients or the server), publicly known and usable templates MAY be created and managed by the server, with a default policy different from private template objects.

Default Operation Policy for Public Template Objects	
Operation	**Policy**
Locate	Allowed to all
Get	Allowed to all
Get Attributes	Allowed to all
Get Attribute List	Allowed to all
Add Attribute	Disallowed to all
Modify Attribute	Disallowed to all
Delete Attribute	Disallowed to all
Destroy	Disallowed to all
Any operation referencing the Template using a Template-Attribute	Allowed to all

Table 65: Default Operation Policy for Public Template Objects

3.29 Link

The *Link* attribute is a structure (see Table 97) used to create a link from one Managed Cryptographic Object to another, closely related target Managed Cryptographic Object. The link has a type, and the allowed types differ, depending on the Object Type of the Managed Cryptographic Object, as listed below. The *Linked Object Identifier* identifies the target Managed Cryptographic Object by its Unique Identifier. The link contains information about the association between the Managed Cryptographic Objects (e.g., the private key corresponding to a public key; the parent certificate for a certificate in a chain; or for a derived symmetric key, the base key from which it was derived).

Possible values of *Link Type* in accordance with the Object Type of the Managed Cryptographic Object are:

- *Private Key Link.* For a Public Key object: the private key corresponding to the public key.
- *Public Key Link.* For a Private Key object: the public key corresponding to the private key. For a Certificate object: the public key contained in the certificate.
- *Certificate Link*. For Certificate objects: the parent certificate for a certificate in a certificate chain. For Public Key objects: the corresponding certificate(s), containing the same public key.
- *Derivation Base Object Link* for a derived Symmetric Key object: the object(s) from which the current symmetric key was derived.
- *Derived Key Link*: the symmetric key(s) that were derived from the current object.
- *Replacement Object Link.* For a Symmetric Key object: the key that resulted from the re-key of the current key. For a Certificate object: the certificate that resulted from the re-certify. Note that there SHALL be only one such replacement object per Managed Object.
- *Replaced Object Link.* For a Symmetric Key object: the key that was re-keyed to obtain the current key. For a Certificate object: the certificate that was re-certified to obtain the current certificate.
- *Wrapping Base Object Link* for an object specified in Encryption Key Information in Key Wrapping Data and used to wrap current object in Get operation. Wrapping Base Object Link SHALL be set before the server performs the actual wrapping of the current object.
- *Wrapped Object Link*: the object that was wrapped as a result of a Get operation using the current object Unique Identifier in the Encryption Key Information in Key Wrapping Data. Wrapped Object Link SHALL be set before the server performs the actual wrapping using the current object.

The Link attribute SHOULD be present for private keys and public keys for which a certificate chain is stored by the server, and for certificates in a certificate chain.

Note that it is possible for a Managed Object to have multiple instances of the Link attribute (e.g., a Private Key has links to the associated certificate, as well as the associated public key; a Certificate object has links to both the public key and to the certificate of the certification authority (CA) that signed the certificate).

It is also possible that a Managed Object does not have links to associated cryptographic objects. This MAY occur in cases where the associated key material is not available to the server or client (e.g., the registration of a CA Signer certificate with a server, where the corresponding private key is held in a different manner).

Object	Encoding	REQUIRED
Link	Structure	
Link Type	Enumeration, see 9.1.3.2.19	Yes
Linked Object	Text String	Yes

Identifier, see 3.1		

Table 97: Link Attribute Structure

SHALL always have a value	No
Initially set by	Client or Server
Modifiable by server	Yes
Modifiable by client	Yes
Deletable by client	Yes
Multiple instances permitted	Yes
When implicitly set	Create Key Pair, Derive Key, Certify, Re-certify, Re-key
Applies to Object Types	All Cryptographic Objects

Table 98: Link Attribute Structure Rules

3.34 Operation Policy Properties

The *Operation Policy Properties* is a structure that provides the client some information about the Operation Policy beyond its name which is captured by the Operation Policy Name Attribute. KMIP v1.1. defines the following elements of the *Operation Policy Properties* structure.

- *Strict Flag.* A Boolean that informs the client if the applicable Operation Policy that applies to the specified objects is secure with respect to possible policy violations that might arise from the use of Derive and Get (wrapped) operations (see [Bjoerkqvist2010] and [Cachin2009] for details). In case the Operation Policy is not secure in the above sense, the Strict Flag SHALL be set to false.
- *Security Level.* One of the four security levels as per **[FIPS140-2]**.

Object	Encoding	REQUIRED
Operation Policy Properties	Structure	
Strict Flag	Boolean	No
Security Level	Enumeration, see 9.1.3.2.31	No

Table 106a: Operation Policy Properties Attribute

SHALL always have a value	No
Initially set by	Server
Modifiable by server	Yes, except that Strict Flag cannot be modified to true
Modifiable by client	No
Deletable by client	No
Multiple instances permitted	No
When implicitly set	Create, Create Key Pair, Register, Derive Key, Re-Key, Re-key Key Pair
Applies to Object Types	All Managed Objects

Table 106b: Operation Policy Properties Attribute Rules

3.35 Owner

The *Owner* attribute designates the Entity who is the owner of the specified object. If a default Operation Policy Name applies to a given object, the Owner SHALL be set to reflect the *creator* of the specified object. For the profiles defined in [KMIP-Prof], the creator SHALL be as defined in [KMIP-Prof].

Object	Encoding	REQUIRED
Owner, see 3.1	Text String	Yes

Table 106c: Owner Attribute

SHALL always have a value	Yes
Initially set by	Server
Modifiable by server	Yes
Modifiable by client	No
Deletable by client	No
Multiple instances permitted	No
When implicitly set	Create, Create Key Pair, Register, Derive Key, Re-Key, Re-key Key Pair, Certify, Re-certify
Applies to Object Types	All Managed Objects

Table 106d: Owner Attribute Rules

4.19 Revoke

This request is used to revoke a Managed Cryptographic Object or an Opaque Object. The request SHALL NOT specify a Template object. The request contains a reason for the revocation (e.g., “key compromise”, “cessation of operation”, etc). Special authentication and authorization SHOULD be enforced to perform this request (see [KMIP-UG]). Only the object Ownercreator or an authorized security officer SHOULD be allowed to issue this request. The operation has one of two effects. If the revocation reason is “key compromise”, then the object is placed into the “compromised” state, and the Compromise Date attribute is set to the current date and time. Otherwise, the object is placed into the “deactivated” state, and the Deactivation Date attribute is set to the current date and time.

Request Payload		
Object	**REQUIRED**	**Description**
Unique Identifier, see 3.1	No	Determines the object being revoked. If omitted, then the ID Placeholder value is used by the server as the Unique Identifier.
Revocation Reason, see 3.26	Yes	Specifies the reason for revocation.
Compromise Occurrence Date, see 3.24	No	SHALL be specified if the Revocation Reason is 'compromised'.

Table 152: Revoke Request Payload

Response Payload		
Object	**REQUIRED**	**Description**
Unique Identifier, see 3.1	Yes	The Unique Identifier of the object.

Table 153: Revoke Response Payload

4.20 Destroy

This request is used to indicate to the server that the key material for the specified Managed Object SHALL be destroyed. The meta-data for the key material MAY be retained by the server (e.g., used to ensure that an expired or revoked private signing key is no longer available). Special authentication and authorization SHOULD be enforced to perform this request (see [KMIP-UG]). Only the object Ownercreator or an authorized security officer SHOULD be allowed to issue this request. If the Unique Identifier specifies a Template object, then the object itself, including all meta-data, SHALL be destroyed. Cryptographic Objects MAY only be destroyed if they are in either Pre-Active or Deactivated state. A Cryptographic Object in the Active state MAY be destroyed if the server sets the Deactivation date (the state of the object transitions to Deactivated) prior to destroying the object.

Request Payload		
Object	**REQUIRED**	**Description**
Unique Identifier, see 3.1	No	Determines the object being destroyed. If omitted, then the ID Placeholder value is used by the server as the Unique Identifier.

Table 154: Destroy Request Payload

Response Payload		
Object	**REQUIRED**	**Description**
Unique Identifier, see 3.1	Yes	The Unique Identifier of the object.

Table 155: Destroy Response Payload

4.21 Archive

This request is used to specify that a Managed Object MAY be archived. The actual time when the object is archived, the location of the archive, or level of archive hierarchy is determined by the policies within the key management system and is not specified by the client. The request contains the unique identifier of the Managed Object. Special authentication and authorization SHOULD be enforced to perform this request (see [KMIP-UG]). Only the object Ownercreator or an authorized security officer SHOULD be allowed to issue this request. This request is only an indication from a client that from its point of view it is possible for the key management system to archive the object.

Request Payload		
Object	**REQUIRED**	**Description**
Unique Identifier, see 3.1	No	Determines the object being archived. If omitted, then the ID Placeholder value is used by the server as the Unique Identifier.

Table 156: Archive Request Payload

Response Payload		
Object	**REQUIRED**	**Description**
Unique Identifier, see 3.1	Yes	The Unique Identifier of the object.

Table 157: Archive Response Payload

9.1.3 Defined Values

This section specifies the values that are defined by this specification. In all cases where an extension mechanism is allowed, this extension mechanism is only able to be used for communication between parties that have pre-agreed understanding of the specific extensions.

9.1.3.1 Tags

The following table defines the tag values for the objects and primitive data values for the protocol messages.

Tag	
Object	**Tag Value**
(Unused)	000000 - 420000
Activation Date	420001
Application Data	420002
Application Namespace	420003
Application Specific Information	420004
Archive Date	420005
Asynchronous Correlation Value	420006
Asynchronous Indicator	420007
Attribute	420008
Attribute Index	420009
Attribute Name	42000A
Attribute Value	42000B
Authentication	42000C
Batch Count	42000D
Batch Error Continuation Option	42000E
Batch Item	42000F
Batch Order Option	420010
Block Cipher Mode	420011
Cancellation Result	420012
Certificate	420013
Certificate Identifier	420014
Certificate Issuer	420015
Certificate Issuer Alternative Name	420016
Certificate Issuer Distinguished Name	420017
Certificate Request	420018

Tag	
Object	**Tag Value**
Certificate Request Type	420019
Certificate Subject	42001A
Certificate Subject Alternative Name	42001B
Certificate Subject Distinguished Name	42001C
Certificate Type	42001D
Certificate Value	42001E
Common Template-Attribute	42001F
Compromise Date	420020
Compromise Occurrence Date	420021
Contact Information	420022
Credential	420023
Credential Type	420024
Credential Value	420025
Criticality Indicator	420026
CRT Coefficient	420027
Cryptographic Algorithm	420028
Cryptographic Domain Parameters	420029
Cryptographic Length	42002A
Cryptographic Parameters	42002B
Cryptographic Usage Mask	42002C
Custom Attribute	42002D
D	42002E
Deactivation Date	42002F
Derivation Data	420030
Derivation Method	420031
Derivation Parameters	420032
Destroy Date	420033
Digest	420034
Digest Value	420035
Encryption Key Information	420036
G	420037
Hashing Algorithm	420038
Initial Date	420039
Initialization Vector	42003A

Tag	
Object	**Tag Value**
Issuer	42003B
Iteration Count	42003C
IV/Counter/Nonce	42003D
J	42003E
Key	42003F
Key Block	420040
Key Compression Type	420041
Key Format Type	420042
Key Material	420043
Key Part Identifier	420044
Key Value	420045
Key Wrapping Data	420046
Key Wrapping Specification	420047
Last Change Date	420048
Lease Time	420049
Link	42004A
Link Type	42004B
Linked Object Identifier	42004C
MAC/Signature	42004D
MAC/Signature Key Information	42004E
Maximum Items	42004F
Maximum Response Size	420050
Message Extension	420051
Modulus	420052
Name	420053
Name Type	420054
Name Value	420055
Object Group	420056
Object Type	420057
Offset	420058
Opaque Data Type	420059
Opaque Data Value	42005A
Opaque Object	42005B
Operation	42005C
Operation Policy Name	42005D

Tag	
Object	**Tag Value**
P	42005E
Padding Method	42005F
Prime Exponent P	420060
Prime Exponent Q	420061
Prime Field Size	420062
Private Exponent	420063
Private Key	420064
Private Key Template-Attribute	420065
Private Key Unique Identifier	420066
Process Start Date	420067
Protect Stop Date	420068
Protocol Version	420069
Protocol Version Major	42006A
Protocol Version Minor	42006B
Public Exponent	42006C
Public Key	42006D
Public Key Template-Attribute	42006E
Public Key Unique Identifier	42006F
Put Function	420070
Q	420071
Q String	420072
Qlength	420073
Query Function	420074
Recommended Curve	420075
Replaced Unique Identifier	420076
Request Header	420077
Request Message	420078
Request Payload	420079
Response Header	42007A
Response Message	42007B
Response Payload	42007C
Result Message	42007D
Result Reason	42007E
Result Status	42007F
Revocation Message	420080
Revocation Reason	420081

Tag	
Object	**Tag Value**
Revocation Reason Code	420082
Key Role Type	420083
Salt	420084
Secret Data	420085
Secret Data Type	420086
Serial Number	420087
Server Information	420088
Split Key	420089
Split Key Method	42008A
Split Key Parts	42008B
Split Key Threshold	42008C
State	42008D
Storage Status Mask	42008E
Symmetric Key	42008F
Template	420090
Template-Attribute	420091
Time Stamp	420092
Unique Batch Item ID	420093
Unique Identifier	420094
Usage Limits	420095
Usage Limits Count	420096
Usage Limits Total	420097
Usage Limits Unit	420098
Username	420099
Validity Date	42009A
Validity Indicator	42009B
Vendor Extension	42009C
Vendor Identification	42009D
Wrapping Method	42009E
X	42009F
Y	4200A0
Password	4200A1
Security Level	4200A2
Strict Flag	4200A3
(Reserved)	4200A4~~2~~ – 42FFFF
(Unused)	430000 – 53FFFF

Tag	
Object	**Tag Value**
Extensions	540000 – 54FFFF
(Unused)	550000 – FFFFFF

Table 193: Tag Values

9.1.3.2.19 Link Type Enumeration

Link Type	
Name	**Value**
Certificate Link	00000101
Public Key Link	00000102
Private Key Link	00000103
Derivation Base Object Link	00000104
Derived Key Link	00000105
Replacement Object Link	00000106
Replaced Object Link	00000107
Wrapping Base Object Link	00000108
Wrapped Object Link	00000109
Extensions	8XXXXXXX

Table 212: Link Type Enumeration

Note: Link Types start at 101 to avoid any confusion with Object Types.

9.1.3.2.31 Security Level Enumeration

Subject Type	
Name	**Value**
Level 1	00000001
Level 2	00000002
Level 3	00000003
Level 4	00000004
Extensions	8XXXXXXX

Table 223a: Security Level Enumeration