Policy and Design Considerations When Using KMIP

(Exclusive of the protocol itself)

This list of considerations is inspired by the protocol, but separate from the protocol itself (e.g., the ID Placeholder is a protocol tool that does not affect the server itself, other than its communication with the client):

Base Objects

Section	Requirement	Key Server	Client
Base Objects			
2 (Objects)	Need to choose the managed objects to be supported.	X	X
2.2.1 (Attribute Object)	Attribute indices are set by the server when a specified named attribute is allowed to have multiple instances., SHALL start with 0, and SHALL never change. Attributes that have a single instance have an Attribute Index of 0.	X	
2.1.2 (Credential Object)	Will a password be/not be/never be required in a credential?	X	X
	Will a credential be used for authentication purposes?	X	
2.1.3 (Key Block Object)	What key format types will be sent/accepted, and how will each be encoded?	X	X
	Will EC keys be expected to be compressed or uncompressed?	X	X
	The Key Block SHALL contain a Key Wrapping Data structure if the key in the Key Value field is wrapped (i.e., encrypted, or MACed/signed, or both).	X	X
2.1.5 (Key Wrapping Data Object)	What key wrapping methods will be allowed?	X	X
	Will MACs or digital signatures or both be allowed, and what algorithms and/or modes and/or padding methods and/or hash functions will be allowed for each (see 9.1.3.2.14 and 9.1.3.2.15)?	X	X
2.1.6 (Key Wrapping Specification)	If Cryptographic Parameters are specified in the Encryption Key Information and/or the MAC/Signature Key Information of the Key Wrapping Specification, then the server SHALL verify that they match one of the instances of the Cryptographic Parameters attribute of the corresponding key. If Cryptographic Parameters are omitted, then the server SHALL use the Cryptographic Parameters attribute with	X	

	the lowest Attribute Index of the corresponding key. If the corresponding key does not have any Cryptographic Parameters attribute, or if no match is found, then an error is returned.		
2.1.7 (Transparent Key Structure)	What format(s) will be allowed for RSA private keys (see Table 15 in 2.1.7.4)?	X	X
	Will a cofactor (J) be allowed for FFDH (see Table 17 in 2.1.7.6)?	X	X
	Will Q and J be expected/disallowed/optional for an FFDH public key (see Table 18 in 2.1.7.7)?	X	X
~~2.1.8 (Template Attribute Structures)~~	~~Will the name and attribute fields be required for a template structure?~~	~~X~~	~~X~~
2.2 (Managed Objects)	Will certificates be handled (2.2.1)?	X	X
	Will symmetric keys be handled (2.2.2)?	X	X
	Will public and/or private keys be handled (2.2.3 and 2.2.4)?	X	X
	Will split keys be handled (2.2.5)? If yes, how many parts (n), and what is the (minimum) threshold (m parts out of a total of n parts)? What methods will be allowed? If primes are used, what are allowable?	X	X
2.2.6 (Templates)	Will templates be allowed? Are there any restrictions on the operations with which they may be used? What objects may be identified in a template?	X	X
2.2.7 (Secret Data)	What kinds of secret data can be handled, if any (see 9.1.3.2.8)? Are key values wrapped?	X	X
2.2.8 (Opaque Objects)	Are custom attributes allowed?	X	X
Attributes			
3 (Attributes)	What attributes (if any) can have multiple instances?	X	X
	Do all instances of an attribute have different values?	X	X
	Which attributes can be updated or deleted? Which are read-only attributes?	X	X
	What is the server policy with regard to returning different attributes for different	X	

	clients?		
	What is the server policy with regard to retaining all, some or none of the object attributes for an object type?	X	
	What is the server policy with regard to the characteristics listed in Table 34?	X	
	Read-only attributes are attributes that SHALL NOT be modified by either server or client, and that SHALL NOT be deleted by a client.	X	X
	A server SHALL NOT delete attributes without receiving a request from a client until the object is destroyed.	X	
	After an object is destroyed, the server MAY retain all, some or none of the object attributes, depending on the object type and server policy. What attributes will be retained?	X	
3.1 (Unique Identifier)	**What is the server policy for generating unique identifiers (i.e., globally unique or REQUIRED to be unique within the identifier space managed by a single key management system)?**	X	
	System expectations for managing unique identifiers are listed in Table 36.	X	
	This attribute SHALL be assigned by the key management system at creation or registration time, and then SHALL NOT be changed or deleted before the object is destroyed. When is the identifier assigned?	X	
3.2 (Name)	What are the rules (if any) for the creation of names by a client? How is the client informed of these rules?		X
	Names SHALL be unique within a given key management domain, but are not REQUIRED to be globally unique. Are the names to be unique within a given key management domain or are they to be globally unique?	X	X
	System expectations for name management are listed in Table 38.	X	X
3.3 (Object Type)	The *Object Type* of a Managed Object (e.g., public key, private key, symmetric key, etc) SHALL be set by the server when the object is created or registered and then SHALL NOT be changed or deleted before the object is destroyed. What object types will be handled (see 9.1.3.2.11)?	X	
3.4 and 3.5	What cryptographic algorithms and key lengths will be	X	X

(Cryptographic Algorithm and Cryptographic Key Length)	supported/available/employed by the server and client (see 9.1.3.2.12 and 9.1.3.2.13)?		
	These attributes SHALL be set by the server when the object is created or registered and then SHALL NOT be changed or deleted before the object is destroyed.	X	
3.6 (Cryptographic parameters)	What cryptographic parameter fields, key types and key role types will be supported(see 9.1.3.2.16)?	X	X
3.7 (Cryptographic Domain Parameters)	Will domain parameters need-to-be/always-to-be/optionally specified in the Create Key Pair Request payload, if this command is handled?	X	X
	Will the NIST Recommended curves be the only ones allowed?	X	X
3.8 (Certificate Type)	The *Certificate Type* value SHALL be set by the server when the certificate is created or registered and then SHALL NOT be changed or deleted before the object is destroyed. What types will be handled (see 9.1.3.2.6) and when will they be set?	X	
3.9 (Certificate Identifier)	The Certificate Identifier SHALL be set by the server when the certificate is created or registered and then SHALL NOT be changed or deleted before the object is destroyed. When is the Certificate Identifier set?	X	
3.10 (Certificate Subject)	How many alternative names (if any) are allowed for a subject?	X	X
	These names SHALL be set by the server, based on the information it extracts from the certificate that is created (as a result of a Certify or a Re-certify operation) or registered (as part of a Register operation) and SHALL NOT be changed or deleted before the object is destroyed.	X	
	If the Subject Alternative Name extension is included in the certificate and is marked *CRITICAL* (i.e., within the certificate itself), then it is possible to issue an X.509 certificate where the subject field is left blank. What will the server do?	X	
3.1 (Certificate Issuer)	How many alternative names (if any) are allowed for an issuer?	X?	?
	The server SHALL set these values based on the information it extracts from a certificate that is created as a result of a Certify or a Re-certify operation or is sent as part of a Register operation. These values SHALL NOT be changed or deleted before the object is destroyed.	X	

3.12 (Digest)	Will multiple digests be computed using different algorithms? Which algorithms will be used?	X	X
	A digest SHALL be computed with the SHA-256 hashing algorithm.	X	X
	The digest(s) SHALL be set by the server when the object is created or registered, provided that the server has access to the Key Material or the Digest Value (possibly obtained via out-of-band mechanisms). If obtain out-of-band, what mechanism will be used?	X	
3.13 (Operation Policy Name)	**What operational policies will be in effect? How are they created and managed, including creation, modification and deletion? When and how will the policy be set? What names will be used?**	X	
	What operations outside of policy control will be allowed (3.13.1)? Which (if any) are allowed at any time?	X	
	When is the policy name attribute set? Is it set explicitly or via some default set by the server?	X	
	A key management system implementation SHALL implement at least one named operation policy (the default 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 (3.13.2); see Tables 62-65 for the default policy. Are the default policies in 3.13 acceptable as the default policies? What other policies will be used?	X	
	Can publicly known and usable templates be created and managed by the server, with a default policy different from private template objects (see 3.13.2.3)?	X	X
3.14 (Cryptographic Usage Mask)	Which masks are appropriate for the server and client? What extensions are allowed?	X	X
	X.509 Key Usage values SHALL be mapped to Cryptographic Usage Mask values in Table 66.	X	X
	Rules for mask management are listed in Table 68.	X	?
3.15 (Lease Time)	The *Lease Time* attribute defines a time interval for a Managed Cryptographic Object beyond which the client SHALL NOT use the object without obtaining another lease. Once its lease expires, the client is only able to renew the lease by calling Obtain		X

	Lease.		
	How will lease times be determined?	X	?
	This attribute always holds the initial length of time allowed for a lease, and not the actual remaining time.	X	X
	Once its lease expires, the client is only able to renew the lease by calling Obtain Lease.		X
	A server SHALL store in this attribute the maximum Lease Time it is able to serve, and a client obtains the lease time (with Obtain Lease) that is less than or equal to the maximum Lease Time. This attribute is read-only for clients. It SHALL be modified by the server only.	X	X
3.16 (Usage Limits)	This attribute only applies to Managed Cryptographic Objects that are able to be used for applying cryptographic protection and it SHALL only reflect their usage for applying that protection (e.g., encryption, signing, etc.).	X	X
	Usage for processing cryptographically-protected data (e.g., decryption, verification, etc.) is not limited.	X	X
	The Usage Limits attribute has the three following fields: • *Usage Limits Total* – the total number of Usage Limits Units allowed to be protected. This is the total value for the entire life of the object and SHALL NOT be changed once the object begins to be used for applying cryptographic protection. • *Usage Limits Count* – the currently remaining number of Usage Limits Units allowed to be protected by the object. • *Usage Limits Unit* – The type of quantity for which this structure specifies a usage limit (e.g., byte, object).	X	X
	When the attribute is initially set (usually during object creation or registration), the Usage Limits Count is set to the Usage Limits Total value allowed for the useful life of the object, and are decremented when the object is used.	X	
	The server SHALL ignore the Usage Limits Count value if the attribute is specified in an operation that creates a new object.	X	
	Changes made via the Modify Attribute operation reflect corrections to the Usage Limits Total value, but they SHALL NOT be changed once the Usage Limits Count value has changed by a Get Usage Allocation operation.	X	

	The Usage Limits Count value SHALL NOT be set or modified by the client via the Add Attribute or Modify Attribute operations.	X	?
	See the attribute rules in the table.	X	X
3.17 (State)	The State SHALL NOT be changed by using the Modify Attribute operation on this attribute.	X	
	The state SHALL only be changed by the server as a part of other operations or other server processes.	X	
	An object SHALL be in one of the following states at any given time: Pre-Active, Active, Deactivated, Compromised, Destroyed, and Destroyed Compromised. See the spec. for what is appropriate for each state and the rules for transitioning from one state to another.	X	
	What is the server policy about transitioning from 1) Deactivated to Destroyed or 2) Compromised to Destroyed Compromised when not explicitly requested by a client?	X	
3.18 (Initial Date)	This attribute SHALL be set by the server when the object is created or registered, and then SHALL NOT be changed or deleted before the object is destroyed.	X	
	This attribute is also set for non-cryptographic objects (e.g., templates) when they are first registered with the server.	X	
3.19 (Activation Date)	This is the date and time when the Managed Cryptographic Object MAY begin to be used.	??	X
	The object SHALL NOT be used for any cryptographic purpose before the *Activation Date* has been reached.	??	X
	Once the state transition from Pre-Active has occurred, then this attribute SHALL NOT be changed or deleted before the object is destroyed.	X	X
	See the attribute rules in the table.	X	X
3.20 (Process Start Date)	This is the date and time when a Managed Symmetric Key Object MAY begin to be used to process cryptographically-protected information (e.g., decryption or unwrapping), depending on the value of its Cryptographic Usage Mask attribute.	??	X
	The object SHALL NOT be used for these cryptographic purposes before the	X	X

	Process Start Date has been reached.		
	This value MAY be equal to or later than, but SHALL NOT preceed, the Activation Date.	X	X
	Once the Process Start Date has occurred, then this attribute SHALL NOT be changed or deleted before the object is destroyed.	X	X
	See the attribute rules in the table	X	X
3.21 (Protect Stop Date)	This is the date and time after which when a Managed Symmetric Key Object SHALL NOT be used for applying cryptographic protection (e.g., encryption or wrapping), depending on the value of its Cryptographic Usage Mask attribute.	X	X
	This value MAY be equal to or earlier than, but SHALL NOT be later than the Deactivation Date.	X	X
	Once the *Protect Stop Date* has occurred, then this attribute SHALL NOT be changed or deleted before the object is destroyed.	X	X
	See the attribute rules in the table.	X	X
3.22 (Deactivation Date)	The *Deactivation Date* is the date and time when the Managed Cryptographic Object SHALL NOT be used for any purpose, except for decryption, signature verification, or unwrapping, but only under extraordinary circumstances and only when special permission is granted. Under what circumstances can the key continue to be used, what permissions are required, and how are they obtained?	X	X
	This attribute SHALL NOT be changed or deleted before the object is destroyed, unless the object is in the Pre-Active or Active state.	X	?
	See the attribute rules in the table.	X	X
3.23 (Destroy Date)	The *Destroy Date* is the date and time when the Managed Object was destroyed. This time corresponds to state transitions 2, 7, or 9 (see Section 3.17).	X	X
	This value is set by the server when the object is destroyed due to the reception of a Destroy operation, or due to server policy or out-of-band administrative action.	X	
	If effected in accordance with server policy, what exactly is this policy, and how is it effected? If set by out-of-band means, what means are used, and how is the destruction effected?	X	X
	See the attribute rules in the table.	X	X

3.24 (Compromise Occurrence Date)	If it is not possible to estimate when the compromise occurred, then this value SHOULD be set to the Initial Date for the object. If not set to the initial date, what date will be used?	X	
3.25 (Compromise Date)	This attribute is set by the server when it receives a Revoke operation with a Revocation Reason of Compromised, or due to server policy or out-of-band administrative action.	X	
	If set by server policy, what is that policy, and how is it effected? If set by out-of-band means, what means are used, and how is the setting of the attribute effected?	X	
	See the attribute rules in the able.	X	X
3.26 (Revocation Reason)	This attribute is only changed by the server as a part of the Revoke Operation.	X	
	Will the Revocation Message field be used/allowed, and how/when will it be used?	X	X
	See the attribute rules in the able.	X	X
3.27 (Archive Date)	This value is set by the server as a part of the Archive operation.	X	
	The server SHALL delete this attribute whenever a Recover operation is performed.	X	
3.28 (Object Group)	An object MAY be part of a group of objects.	X	
	An object MAY belong to more than one group of objects.	X	
	To assign an object to a group of objects, the object group name SHOULD be set into this attribute.		
	Can object groups be used? If used, can an object belong to more than one group? How are the groups named?	X	X
	See the attribute rules in the table.	X	X
3.29 (Link)	What link types are allowed (private key, public key, certificate, derivation base object, derived key, replacement object,replaced object)? How are they named, and under what conditions are they used?	X	X
	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.	X	X
	See the attribute rules in the table.	X	X
3.30 (Application-Specific	Clients MAY request to set (i.e., using any of the operations that result in new Managed Object(s) on the server or adding/modifying the attribute of an existing Managed Object) an instance of this attribute with a particular Application		X

Information)	Namespace while omitting Application Data.		
	The *Application Specific Information* attribute consists of the following fields: an *Application Namespace* and *Application Data* specific to that application namespace.	X	X
	Will clients be allowed to request to set this information (i.e., using any of the operations that result in new Managed Object(s) on the server or add/modify the attribute of an existing Managed Object) an instance of this attribute with a particular Application Namespace while omitting Application Data? In this case, if the server supports this namespace (as indicated by the Query operation in Section **Error! Reference source not found.**), then it SHALL return a suitable Application Data value. If the server does not support this namespace, then an error SHALL be returned.	X	X
	See the attribute rules in the table.	X	X
3.31 (Contact Information)	What contact information (if any) will be available (to the protocol)?	X	?
	See the attribute rules in the table.	X	X
3.32 (Last Change Date)	A required field.	X	?
	See the attributes in the table.	X	X
3.33 (Custom Attribute)	Created by the client and not interpreted by the server, or is created by the server and MAY be interpreted by the client.	X	X
	Will custom attributes be allowed?	X	X
	All custom attributes created by the client SHALL adhere to a naming scheme, where the name of the attribute SHALL have a prefix of 'x-'.		X
	All custom attributes created by the key management server SHALL adhere to a naming scheme where the name of the attribute SHALL have a prefix of 'y-'.	X	
	The server SHALL NOT accept a client-created or modified attribute, where the name of the attribute has a prefix of ‘y-‘.	X	
	See the attribute rules in the table.	X	X
Client to Server Operations			
4 (Client to Server Operations)	What operations can be requested by a client (see 9.1.3.2.26)?	?	X
	Any client that issues a specific request SHALL be capable of understanding the response to the request.		X

	Multiple operations MAY be combined within a batch, resulting in a single request/response message pair.	X	X
	Client requests MAY contain attribute values to be assigned to the object.	?	X
	Server responses MAY contain attribute values that were not specified in the request, but have been implicitly set by the server. How is these attribute values generated and recorded by the server?	X	?
	For any operations that operate on Managed Objects already stored on the server, any archived object SHALL first be made available by a Recover operation before they MAY be specified (i.e., as on-line objects).	X	X
4.1 (Create)	Can the server create symmetric keys?	X	
	The request contains information about the type of object being created, and some of the attributes to be assigned to the object This information MAY be specified by the names of Template objects that already exist. Can this operation use template objects?	X	X
4.2 (Create Key Pair)	Can the server create new key pairs? If, so, they SHALL be registered. What does registration consist of?	X	?
	The server SHALL create a link between the two key pair objects.	X	?
	Unique identifiers SHALL be created for each key of the key pair.		
4.3 (Register)	If implemented, the server SHALL register an object upon request by the client.	X	
	Does the client create objects, obtain objects by some other means, or both? If obtained by other means, how are they obtained and from where?		X
	What objects are to be stored by the server, stored only by the client, or both?	X	X
	The server SHALL generate a Unique Identifier to the registered object.	X	
	The Initial Date attribute of the object SHALL be set to the current time.	X	
4.4 (Re-Key)	Attributes of the replacement key SHALL be copied from the existing key, with the exception of the attributes listed in the tables.	X	
	As the replacement key takes over the name attribute of the existing key, Re-key SHOULD only be performed once on a given key. What is the server policy for allowing multiple rekeys? Is it enforced?	X	
	As a result of Re-key, the Link attribute of the existing key is set to point to the replacement key and vice versa.	X	

4.5 (Derived Key)	This request is used to derive a symmetric key or Secret Data object from a key or secret data that is already known to the key management system (see 9.1.3.2.20). The request SHALL only apply to Managed Cryptographic Objects that have the Derive Key bit set in the Cryptographic Usage Mask attribute (9.1.3.3.1) of the specified Managed Object (i.e., are able to be used for key derivation).	X	X
	The client SHALL specify the desired length of the derived key or Secret Data object and the Cryptographic Algorithm. If the specified length exceeds the output of the derivation method, then the server SHALL return an error.	X	X
	Is the client allowed to request the creation of multiple keys and IVs by requesting the creation of a Secret Data object and specifying a Cryptographic Length that is the total length of the derived object? If so, the length SHALL NOT exceed the length of the output returned by the chosen derivation method.	X	X
	Which key derivation methods are allowed (see 4.5)?	X	X
	When requested, the server SHALL perform the derivation function, and then register the derived object as a new Managed Object, returning the new Unique Identifier for the new object in the response.	X	
	The Link attributes (i.e., Derived Key Link in the objects from which the key is derived, and the Derivation Base Object Link in the derived key) of all objects involved SHALL be set to point to the corresponding objects.	X	
	If a key is to be derived using the HASH derivation method, then clients are REQUIRED to indicate the hash algorithm.		X
	If a key is to be derived using the CBC mode, then clients are REQUIRED to indicate the Block Cipher Mode).		X
	The server SHALL verify that the specified mode matches one of the instances of Cryptographic Parameters set for the corresponding key. If Cryptographic Parameters are omitted, then the server SHALL select the Cryptographic Parameters with the lowest Attribute Index for the specified key. If the corresponding key does not have any Cryptographic Parameters attribute, or if no match is found, then an error is returned.	X	
	For the NIST SP 800-108 methods**Error! Reference source not found.**, Derivation Data is Label||{0x00}||Context, where the all-zero byte is OPTIONAL	X	
	Derivation data MAY either be explicitly provided by the client with the Derivation Data field or implicitly provided by providing the Unique Identifier of a Secret Data object. If both are provided, then an error SHALL be returned.	X	X

4.6 (Certify)	For the certify operation to be handled, the server SHALL be or have access to a CA.	X	?
	What types of certificates can be requested?	X	
	As a result of Certify, the Link attribute of the Public Key and of the generated certificate SHALL be set to point at each other.	X	
4.7 (Re-certify)	For the re-certify operation to be handled, the server SHALL be or have access to a CA. What types of certificates can be requested?	X	?
	The new certificate takes over the name attribute of the existing certificate; Re-certify SHOULD only be performed once on a given (existing) certificate. What is the server policy with respect to handling this "SHOULD"? Are any restrictions enforced?	X	
	The Link attribute of the existing certificate and of the new certificate are set to point at each other.	X	
	The Link attribute of the Public Key is changed to point to the new certificate.	X	
	Does the server handle offsets? If Offset is set, then the dates of the new certificate SHALL be set based on the dates of the existing certificate (if such dates exist) as specified in Table 126. Attributes that are handled differently are addressed in Table 127.	X	X
4.8 (Locate)	Can attribute indexes in the request be handled?	X	
	The request MAY contain a *Maximum Items* field, which specifies the maximum number of objects to be returned. If the Maximum Items field is omitted, will the server return all objects matched, or impose an internal maximum limit due to resource limitations, or something else?	X	X
	If more than one object satisfies the identification criteria specified in the request, will the response contain Unique Identifiers for multiple Managed Objects?	X	
	Returned objects SHALL match **all** of the attributes in the request.	X	
	If no attribute is specified in the request, any object SHALL be deemed to match the Locate request.	X	
	Are wild-cards or regular expressions (defined, e.g., in [ISO/IEC 9945-2]) supported by the server for matching attribute fields when the field type is a Text String or a Byte String?	X	?
	If a single instance of a given Date attribute is used in the request (e.g., the	X	

	Activation Date), then objects with the same Date attribute are considered to be matching candidate objects.		
	If two instances of the same Date attribute are used (i.e., with two different values specifying a range), then objects for which the Date attribute is inside or at a limit of the range are considered to be matching candidate objects.	X	
	If a Date attribute is set to its largest possible value, then it is equivalent to an undefined attribute.	X	
	When the Cryptographic Usage Mask attribute is specified in the request, candidate objects are compared against this field via an operation that consists of a logical AND of the requested mask with the mask in the candidate object, and then a comparison of the resulting value with the requested mask.	X	
	When the Usage Allocation attribute is specified in the request, matching candidate objects SHALL have an Object or Byte Count and Total Objects or Bytes equal to or larger than the values specified in the request.	X	
	When an attribute that is defined as a structure is specified, all of the structure fields are not REQUIRED to be specified.	X	
	The Storage Status Mask field is used to indicate whether only on-line objects, only archived objects, or both on-line and archived objects are to be searched.	X	
	The server MAY store attributes of archived objects in order to expedite Locate operations that search through archived objects; will the server do so?	X	
4.9 (Check)	If the server determines that the client is allowed to use the object according to the specified attributes, then the server returns the Unique Identifier of the object.	X	
	If the server determines that the client is not allowed to use the object according to the specified attributes, then the server does not return the Unique Identifier, and the operation returns the set of attributes specified in the request that caused the server policy denial.	X	
	The cryptographic usage mask is used to specify the cryptographic operations for which the client intends to use the object. This allows the server to determine if the policy allows this client to perform these operations with the object.	X	X
4.10 (Get)	Will the server wrap returned keys: always? never? as requested by the client? Depending on the key and type of key?	X	
	If a client registered a key in a given format, the server SHALL be able to return the	X	

	key during the Get operation in the same format that was used when the key was registered.		
	What other format conversions are supported by the server?	X	
4.11 (Get Attributes)	If a specified attribute has multiple instances, then all instances SHALL be returned.	X	
	If no requested attributes exist, then the response SHALL consist only of the Unique Identifier.	X	
	If no attribute name is specified in the request, all attributes SHALL be deemed to match the Get Attributes request.	X	
4.13 (Add Attribute)	For non-multi-instance attributes, this is how the attribute value is created.	X	X
	For multi-instance attributes, this is how the first and subsequent values are created.	X	X
	Existing attribute values SHALL only be changed by the Modify Attribute operation.	X	X
	Read-Only attributes SHALL NOT be added using the Add Attribute operation.	X	?
	Create a new attribute index; indices begin at 0.	X	
4.14 (Modify Attribute)	Only existing attributes MAY be changed via this operation.	X	?
	If an Attribute Index is specified, then only the specified instance of the attribute is modified.	X	?
	If the attribute has multiple instances, and no Attribute Index is specified in the request, then the Attribute Index SHALL be assumed to be 0.	X	?
4.15 (Delete Attribute)	Attributes that SHALL always have a value SHALL never be deleted by this operation.	X	?
	If no Attribute Index is specified, and the Attribute whose name is specified has multiple instances, then the operation is rejected.	X	
	Attempting to delete a non-existent attribute or specifying an Attribute Index for which there exists no Attribute Value SHALL result in an error.	X	
	Only a single attribute instance SHALL be deleted at a time. Multiple delete operations (e.g., possibly batched) are necessary to delete several attribute instances.	X	X
	Attempting to delete a non-existent attribute or specifying an Attribute Index for which there exists no Attribute Value SHALL result in an error.	X	

4.16 (Object Lease)	The server SHALL return a lease time of zero, to indicate that no lease interval is effective; in this case, the client MAY use the object without any lease time limit.	X	X
	If a client's lease expires, then the client SHALL NOT use the associated cryptographic object until a new lease is obtained.		X
	If the server determines that a new lease SHALL NOT be issued for the specified cryptographic object, then the server SHALL respond to the Obtain Lease request with an error.	X	
	The current value of the Last Change Date attribute MAY be used by the client to determine if any of the attributes cached by the client need to be refreshed, by comparing this time to the time when the attributes were previously obtained.		X
4.17 (Get Usage Allocation)	The allocation only applies to Managed Cryptographic Objects that are able to be used for applying protection (e.g., symmetric keys for encryption, private keys for signing, etc.) and is only valid if the Managed Cryptographic Object has a Usage Limits attribute.	X	
	Usage for processing cryptographically-protected information (e.g., decryption, verification, etc.) is not limited and SHALL not be allocated.	X	
	A Managed Cryptographic Object that has a Usage Limits attribute SHALL NOT be used by a client for applying cryptographic protection unless an allocation has been obtained using this operation.		X
	If the operation is requested for an object that has no Usage Limits attribute, or is not an object that MAY be used for applying cryptographic protection, then the server SHALL return an error.	X	
	The field in the request specifies the number of units that the client needs to protect. If the requested amount is not available or if the Managed Object is not able to be used for applying cryptographic protection at this time, then the server SHALL return an error.	X	?
	The server SHALL assume that the entire allocated amount is going to be consumed.	X	
	Once the entire allocated amount has been consumed, the client SHALL NOT continue to use the Managed Cryptographic Object for applying cryptographic protection until a new allocation is obtained.		X
4.18 (Activate)	The operation SHALL only be performed on an object in the Pre-Active state. The server SHALL change the state to Active, and set the Activation Date to the current date and time.	X	?

4.19 (Revoke)	**Special authentication and authorization SHOULD be enforced to perform this request. Is authentication and authentication enforced? If, so, how?**	X	
	Only the object creator or an authorized security officer SHOULD be allowed to issue this request. What is the server policy with respect to this, and how is it enforced?	X	
	If the revocation reason is “compromised”, then the object SHALL be placed into the “compromised” state, and the Compromise Date attribute SHALL be set to the current date and time. Otherwise, the object SHALL be placed into the “deactivated” state, and the Deactivation Date attribute SHALL be set to the current date and time.	X	
4.20 (Destroy)	Is any of the meta-data for the key material retained by the server?	X	
	Does the server enforce special authentication and authorization to perform this request?	X	
	Only the object creator or an authorized security officer SHOULD be allowed to issue this request. What is the server policy on this, is it enforced, and how?	X	?
	If the Unique Identifier specifies a Template object, then the object itself, including all meta-data, SHALL be destroyed.	X	
	Cryptographic Objects SHALL only be destroyed if they are in either Pre-Active or Deactivated state.	X	
	If the key material to be destroyed is in the Active state, the server SHALL deactivate the key prior to destroying the object.	X	
4.21 (Archive)	What managed objects are archived?	X	
	What is the server policy with regard to the time of archiving an object, the location of the archive, and the level of archive hierarchy?		
	How is authorization enforced (if at all)?	X	
	Does the server enforce special authentication and authorization to perform this request?	X	
	Only the object creator or an authorized security officer SHOULD be allowed to issue this request. What is the server policy with respect to who can archive an object? Is the policy enforced, and how?	X	?
4.22 (Recover)	Is asynchronous polling required to obtain the response due to delays caused by retrieving the object from the archive?	?	?
	Does the server enforce special authentication and authorization to perform this	X	

	request?		
4.23 (Validate)	Does the server support this operation?	X	
	Is a date required for which all certificates in the certificate chain are REQUIRED to be valid?	X	X
	What method or policy is the used by the server to conduct a validation?	X	
	In what order does the server validate the supplied certificate chain and the specification of trust anchors used to terminate validation?	X	
4.24 (Query)	Can unauthenticated clients invoke a query to interrogate server features and functions?	X	?
	The *Operation* fields in the response SHALL contain Operation enumerated values that list all the operations that the server supports.	X	X
	The *Object Type* fields in the response SHALL contain Object Type enumerated values that list all the object types that the server supports.	X	
	The server SHALL generate values as indicated in the Application Namespace fields in the request.	X	
	If the response payload is empty, the server SHALL not return any values.	X	
4,25 (Cancel)	The server SHALL respond with a *Cancellation Result* that contains one of the values listed in 4.25.	X	
	The response to this operation SHALL not be asynchronous.	X	
4.26 (Poll)	The response to this operation SHALL NOT be asynchronous.	X	X
	If the operation has completed, the response SHALL contain the appropriate payload for the operation. This response SHALL be identical to the response that would have been sent if the operation had completed synchronously.	X	
Server to Client Operations			
5.1 (Notify)	This operation is only ever sent by a server to a client via means outside of the normal client request/response protocol, using information known to the server via unspecified configuration or administrative mechanisms. What are the configuration or administrative mechanisms to be used?	X	?
	The client SHALL send a response in the form of a Response Message .containing	?	X

	no payload, unless both the client and server have prior knowledge (obtained via out-of-band mechanisms) that the client is not able to respond. What out-of-band mechanism will be used?		
5.2 (Put)	This operation is only ever sent by a server to a client via means outside of the normal client request/response protocol, using information known to the server via unspecified configuration or administrative mechanisms. What are the configuration or administrative mechanisms to be used?	X	?
	The client SHALL send a response in the form of a Response Message containing no payload, unless both the client and server have prior knowledge (obtained via out-of-band mechanisms) that the client is not able to respond. What out-of-band mechanism will be used?	?	X
	If a REPLACE value is sent, and the Replaced Unique Identifier does not exist at the client, the client SHALL interpret this as if the Put Function contained the value New.		X
	The server MAY include attributes with the object to specify how the object is to be used by the client. The server MAY include a Lease Time attribute that grants a lease to the client. Can the client handle this?	X	X
	If the Managed Object is a wrapped key, then the key wrapping specification SHALL be exchanged prior to the transfer via out-of-band mechanisms. What out-of-band mechanism will be used?	X	X
Message Contents	Content to be handled by an implementation depends on the operations that could be used.		
6.5 (Time Stamp)	It is used for time stamping, and MAY be used to enforce reasonable time usage at a client (e.g., a server MAY choose to reject a request if a client's time stamp contains a value that is too far off the server's time).	X	X
6.6 (Authentication)	**Servers MAY require authentication on no requests, a subset of the requests, or all requests, depending on policy. What is the server policy on this?**	X	?
	Query operations used to interrogate server features and functions SHOULD NOT require authentication. What is the server's policy on authentication for a query operation?	X	X
6.16 (Message Extensions)	What message extensions (if any) are allowed?	X	X
Server Baseline			

12.1 (KMIP Server)	Minimal KMIP conformance is specified:	X	

Interesting definitions:

Archive	To place information not accessed frequently into long-term storage.
Recover	To retrieve information that was archived to long-term storage.