2.1.2 Credential

A *Credential* is a structure (see Table 3~~Table 3~~) used for client identification purposes and is not managed by the key management system (e.g., user id/password pairs, Kerberos tokens, etc). It MAY be used for authentication purposes as indicated in **[KMIP-Prof]**.

Object	Encoding	REQUIRED
Credential	Structure	
~~Credential~~ Subject Type	Enumeration, see 9.1.3.2.1	Yes
Subject Value	Varies according to Subject Type	No
Subject Authentication Information Type	Enumeration, see 9.1.3.2.2	Yes
~~Credential~~ Subject Authentication Information Value	~~Varies. Structure for Username and Password Credential Type.~~Varies according to Subject Authentication Information Type	~~Yes~~No

Table 3: Credential Object Structure

~~If the Credential Type in the Credential is *Username and Password*, then Credential Value is a structure as shown in Table 4. The Username field identifies the client, and the Password field is a secret that authenticates the client.~~

~~Object~~	~~Encoding~~	~~REQUIRED~~
~~Credential Value~~	~~Structure~~	
~~Username~~	~~Text String~~	~~Yes~~
~~Password~~	~~Text String~~	~~No~~

~~**Table 4: Credential Value Structure for the Username and Password Credential**~~

The following table describes the Subject Value encodings for different values of Subject Type.

Subject Type	Encoding
Username	Text String
X.509 Certificate Identifier	Structure, See section 3.9

The following table describes the Subject Authentication Information Value encodings for different values of Subject Authentication Information Type.

Formatted: Normal

Copyright © OASIS® 2010. All Rights Reserved.

Subject Authentication Information Type	Encoding
Password	Text String
X.509 Certificate	Text String

2.1.3 Key Block

A *Key Block* object is a structure (see Table 4~~Table 5~~) used to encapsulate all of the information that is closely associated with a cryptographic key. It contains a Key Value of one of the following *Key Format Types*:

- *Raw* – This is a key that contains only cryptographic key material, encoded as a string of bytes.
- *Opaque* – This is an encoded key for which the encoding is unknown to the key management system. It is encoded as a string of bytes.
- *PKCS1* – This is an encoded private key, expressed as a DER-encoded ASN.1 PKCS#1 object.
- *PKCS8* – This is an encoded private key, expressed as a DER-encoded ASN.1 PKCS#8 object, supporting both the RSAPrivateKey syntax and EncryptedPrivateKey.
- *X*.509 – This is an encoded object, expressed as a DER-encoded ASN.1 X.509 object.
- ECPrivateKey – This is an ASN.1 encoded elliptic curve private key.
- Several *Transparent Key* types – These are algorithm-specific structures containing defined values for the various key types, as defined in Section 2.1.7
- *Extensions* – These are vendor-specific extensions to allow for proprietary or legacy key formats.

The Key Block MAY contain the Key Compression Type, which indicates the format of the elliptic curve public key. By default, the public key is uncompressed.

The Key Block also has the Cryptographic Algorithm and the Cryptographic Length of the key contained in the Key Value field. Some example values are:

- RSA keys are typically 1024, 2048 or 3072 bits in length
- 3DES keys are typically from 112 to 192 bits (depending upon key length and the presence of parity bits)
- AES keys are 128, 192 or 256 bits in length

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).

Copyright © OASIS® 2010. All Rights Reserved.

The inverse of an element $uy + v$ is given by

$(uy + v)^{-1} = ud^{-1}y + (u + v)d^{-1}$, where $d = (u + v)v + mu^2$.

2.2.6 Template

A *Template* is a named Managed Object containing the client-settable attributes of a Managed Cryptographic Object (i.e., a stored, named list of attributes). A Template is used to specify the attributes of a new Managed Cryptographic Object in various operations. It is intended to be used to specify the cryptographic attributes of new objects in a standardized or convenient way. None of the client-settable attributes specified in a Template except the Name attribute apply to the template object itself, but instead apply to any object created using the Template.

The Template MAY be the subject of the Register, Locate, Get, Get Attributes, Get Attribute List, Add Attribute, Modify Attribute, Delete Attribute, and Destroy operations.

An attribute specified in a Template is applicable either to the Template itself or to objects created using the Template.

Attributes applicable to the Template itself are: Unique Identifier, Object Type, Name, Initial Date, Archive Date, and Last Change Date.

Attributes applicable to objects created using the Template are:

- Cryptographic Algorithm
- Cryptographic Length
- Cryptographic Domain Parameters
- Cryptographic Parameters
- Operation Policy Name
- Cryptographic Usage Mask
- Usage Limits
- Activation Date
- Process Start Date
- Protect Stop Date
- Deactivation Date
- Object Group
- Application Specific Information
- Contact Information
- Custom Attribute

Object	Encoding	REQUIRED
Template	Structure	
Attribute	Attribute Object, see 2.1.1	Yes. MAY be repeated.

Table 3031: Template Object Structure

Copyright © OASIS® 2010. All Rights Reserved.

2.2.7 Secret Data

A Managed Cryptographic Object containing a shared secret value that is not a key or certificate (e.g., a password). The Key Block of the *Secret Data* object contains a Key Value of the Opaque type. The Key Value MAY be wrapped.

Object	Encoding	REQUIRED
Secret Data	Structure	
Secret Data Type	Enumeration, see 9.1.3.2.99~~.1.3.2.8~~	Yes
Key Block	Structure, see 2.1.3	Yes

Table 31~~32~~: Secret Data Object Structure

2.2.8 Opaque Object

A Managed Object that the key management server is possibly not able to interpret. The context information for this object MAY be stored and retrieved using Custom Attributes.

Object	Encoding	REQUIRED
Opaque Object	Structure	
Opaque Data Type	Enumeration, see 9.1.3.2.109~~.1.3.2.9~~	Yes
Opaque Data Value	Byte String	Yes

Table 32~~33~~: Opaque Object Structure

2.2.9 Entity

A Managed Object that represents a client of the key management server. Entity objects consist of one or more Credential objects that contain a Subject identifying the client and optional Subject Authentication Information that is used to authenticate the Entity. Subjects SHALL be unique within a given key management domain, but are not REQUIRED to be globally unique.

Object	Encoding	REQUIRED
Entity	Structure	
Credential	Structure, see 2.1.2	Yes, May be repeated

Table 33a: Entity Structure

Formatted: Normal

Copyright © OASIS® 2010. All Rights Reserved.

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 controlled by Entity Operation Policy~~outside of operation policy control~~

The following operations are controlled by an Entity Operation Policy (See section 3.35) associated with an Entity object:~~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~~owner 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.

Copyright © OASIS® 2010. All Rights Reserved.

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

Copyright © OASIS® 2010. All Rights Reserved.

Activate	Allowed to ~~creator~~owner only
Revoke	Allowed to ~~creator~~owner only
Destroy	Allowed to ~~creator~~owner only
Archive	Allowed to ~~creator~~owner only
Recover	Allowed to ~~creator~~owner 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 ~~creator~~owner only
Get	Allowed to ~~creator~~owner only
Get Attributes	Allowed to ~~creator~~owner only
Get Attribute List	Allowed to ~~creator~~owner only
Add Attribute	Allowed to ~~creator~~owner only
Modify Attribute	Allowed to ~~creator~~owner only
Delete Attribute	Allowed to ~~creator~~owner only
Destroy	Allowed to ~~creator~~owner only
Any operation referencing the Template using a Template-Attribute	Allowed to ~~creator~~owner 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

Copyright © OASIS® 2010. All Rights Reserved.

Any operation referencing the Template using a Template-Attribute	Allowed to all

Table 65: Default Operation Policy for Public Template Objects

3.13.2.4 Default Operation Policy for Entity Objects

This policy applies to Entity objects.

Default Operation Policy for Entity Objects	
Operation	**Policy**
Locate	Allowed to All
Get	Allowed to Entity only
Get Attributes	Allowed to All
Get Attribute List	Allowed to All
Add Attribute	Allowed to Entity only
Modify Attribute	Allowed to Entity only
Delete Attribute	Allowed to Entity only
Destroy	Allowed to Entity only

Table 65a: Default Operation Policy for Entity Objects

Formatted: Normal

3.14 Cryptographic Usage Mask

The *Cryptographic Usage Mask* defines the cryptographic usage of a key. This is a bit mask that indicates to the client which cryptographic functions MAY be performed using the key, and which ones SHALL NOT be performed.

- Sign
- Verify
- Encrypt
- Decrypt
- Wrap Key
- Unwrap Key
- Export
- MAC Generate
- MAC Verify
- Derive Key
- Content Commitment
- Key Agreement
- Certificate Sign
- CRL Sign
- Generate Cryptogram
- Validate Cryptogram
- Translate Encrypt
- Translate Decrypt
- Translate Wrap
- Translate Unwrap

Copyright © OASIS® 2010. All Rights Reserved.

SHALL always have a value	No
Initially set by	Server
Modifiable by server	No
Modifiable by client	No
Deletable by client	No
Multiple instances permitted	No
When implicitly set	Destroy
Applies to Object Types	All Cryptographic Objects, Opaque Objects, Entity Objects

Table 86: Destroy Date Attribute Rules

3.24 Compromise Occurrence Date

The *Compromise Occurrence Date* is the date and time when the Managed Cryptographic Object was first believed to be compromised. If it is not possible to estimate when the compromise occurred, then this value SHOULD be set to the Initial Date for the object.

Object	Encoding	
Compromise Occurrence Date	Date-Time	

Table 87: Compromise Occurrence Date Attribute

SHALL always have a value	No
Initially set by	Server
Modifiable by server	No
Modifiable by client	No
Deletable by client	No
Multiple instances permitted	No
When implicitly set	Revoke
Applies to Object Types	All Cryptographic Objects, Opaque Object

Table 88: Compromise Occurrence Date Attribute Rules

3.25 Compromise Date

The *Compromise Date* is the date and time when the Managed Cryptographic Object entered into the compromised state. This time corresponds to state transitions 3, 5, 8, or 10 (see Section 3.17). This time indicates when the key management system was made aware of the compromise, not necessarily when the compromise occurred. 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.

Object	Encoding	
Compromise Date	Date-Time	

Table 89: Compromise Date Attribute

Copyright © OASIS® 2010. All Rights Reserved.

Object	Encoding	
Archive Date	Date-Time	

Table 93: Archive Date Attribute

SHALL always have a value	No
Initially set by	Server
Modifiable by server	No
Modifiable by client	No
Deletable by client	No
Multiple instances permitted	No
When implicitly set	Archive
Applies to Object Types	All Cryptographic Objects, Opaque Object~~All Objects~~

Table 94: Archive Date Attribute Rules

3.28 Object Group

An object MAY be part of a group of objects. An object MAY belong to more than one group of objects. To assign an object to a group of objects, the object group name SHOULD be set into this attribute.

Object	Encoding	
Object Group	Text String	

Table 95: Object Group Attribute

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, Create Key Pair, Register, Derive Key, Certify, Re-certify, Re-key
Applies to Object Types	All Objects

Table 96: Object Group Attribute Rules

3.29 Link

The *Link* attribute is a structure (see Table 97~~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).

Copyright © OASIS® 2010. All Rights Reserved.

a prefix of 'y-'. The tag type Custom Attribute is not able to identify the particular attribute; hence such an attribute SHALL only appear in an Attribute Structure with its name as defined in Section 2.1.1.

Object	Encoding	
Custom Attribute	Any data type or structure. If a structure, then the structure SHALL NOT include sub structures	The name of the attribute SHALL start with 'x-' or 'y-'.

Table 105 Custom Attribute

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

Table 106: Custom Attribute Rules

3.34 Owner

The Owner attribute specifies the Unique Identifier of the Entity object that owns the specified object. By default, it is set to the unique identifier of the Entity object that created the object. A server MAY allow a client to specify an Object Owner other than the client itself dependent on server policy.

Object	Encoding	
Owner	Text String	

Table 105a Owner attribute encoding

SHALL always have a value	Yes
Initially set by	Client or Server
Modifiable by server	Yes
Modifiable by client	Yes
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 Cryptographic Objects,

Copyright © OASIS® 2010. All Rights Reserved.

	Templates, Opaque Objects

Table 105b: Custom Attribute Rules

3.35 Entity Operation Policy Name

An entity operation policy controls which key management operations an entity MAY perform. The content of the Entity 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. This policy governs operations that do not operate on any object. The Entity Operation Policy Name attribute SHOULD be set when an Entity object is created or registered. It is set either explicitly or via some default set by the server, which then applies the named policy to all subsequent operations requested by the entity.

Object	Encoding	
Entity Operation Policy Name	Text String	

Table 107: Contact Information Attribute

SHALL always have a value	No
Initially set by	Client or 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, Certify, Re-certify, Re-key
Applies to Object Types	Entity

Table 105c: Entity Operation Policy Attribute Rules

3.35.1 Default Entity Operation Policy

A key management system implementation SHALL implement at least one named entity operation policy, which is used for entities that do not have this attribute set. This policy is named *entity-default*. It specifies the following rules for key management operations:

Formatted: Font: Bold
Formatted: Normal, Centered
Formatted: Font: Bold
Formatted: Normal
Formatted: Font: Bold
Formatted: Normal, Centered
Formatted: Font: Bold
Formatted: Font: Bold
Formatted: Normal
Formatted: Font: Italic

Copyright © OASIS® 2010. All Rights Reserved.

Default Operation Policy for Entity Objects		
Operation	**Object Type**	**Policy**
Create	Symmetric Key	Allowed to all
Create Key Pair	Public Key, Private Key	Allowed to all
Register	All, except Entity	Allowed to all
Certify	Public Key	Allowed to all
Re-certify	Certificate	Allowed to all
Validate	Certificate	Allowed to all
Query	N/A	Allowed to all
Cancel	N/A	Allowed to all
Poll	N/A	Allowed to all

991 **Table105d: Default Operation Policy for Entity Objects**

992

Formatted: Normal

Copyright © OASIS® 2010. All Rights Reserved.

Attribute	REQUIRED	SHALL contain the same value for both Private and Public Key
Cryptographic Algorithm, see 3.4	Yes	Yes
Cryptographic Length, see 3.5	No	Yes
Cryptographic Usage Mask, see 3.14	Yes	No
Cryptographic Domain Parameters, see 3.7	No	Yes
Cryptographic Parameters, see 3.6	No	Yes

Table 113~~112~~: Create Key Pair Attribute Requirements

Setting the same Cryptographic Length value for both private and public key does not imply that both keys are of equal length. For RSA, Cryptographic Length corresponds to the bit length of the Modulus. For DSA and DH algorithms, Cryptographic Length corresponds to the bit length of parameter P, and the bit length of Q is set separately in the Cryptographic Domain Parameters attribute. For ECDSA, ECDH, and ECMQV algorithms, Cryptographic Length corresponds to the bit length of parameter Q.

4.3 Register

This operation requests the server to register a Managed Object that was created by the client or obtained by the client through some other means, allowing the server to manage the object. The arguments in the request are similar to those in the Create operation, but also MAY contain the object itself for storage by the server. Optionally, objects that are not to be stored by the key management system MAY be omitted from the request (e.g., private keys).

The request contains information about the type of object being registered and some of the attributes to be assigned to the object (e.g., Cryptographic Algorithm, Cryptographic Length, etc). This information MAY be specified by the use of a Template-Attribute object.

Some key management servers MAY require a Security Officer to approve registration of an Entity. This request MAY require asynchronous polling to obtain the response.

The response contains the Unique Identifier assigned by the server to the registered object. The server SHALL copy the Unique Identifier returned by this operations into the ID Placeholder variable. The Initial Date attribute of the object SHALL be set to the current time.

Request Payload		
Object	**REQUIRED**	**Description**
Object Type, see 3.3	Yes	Determines the type of object being registered.
Template-Attribute, see 2.1.8	Yes	Specifies desired object attributes using templates and/or individual attributes.
Certificate, Symmetric Key, Private Key, Public Key, Split Key, Template Secret Data~~,~~ ~~or~~ Opaque Object or Entity, see 2.2	No	The object being registered. The object and attributes MAY be wrapped. Some objects (e.g., Private Keys), MAY be omitted from the request.

Table 114~~113~~: Register Request Payload

Copyright © OASIS® 2010. All Rights Reserved.

4.18 Activate

This request is used to activate a Managed Cryptographic Object. The request SHALL NOT specify a Template object. The operation SHALL only be performed on an object in the Pre-Active state and has the effect of changing its state to Active, and setting its Activation Date to the current date and time.

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

Table 151~~150~~: Activate Request Payload

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

Table 152~~151~~: Activate Response Payload

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 ~~creator~~8 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 153~~152~~: Revoke Request Payload

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

Table 154~~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

Copyright © OASIS® 2010. All Rights Reserved.

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 ~~creator~~owner 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 155~~154~~: Destroy Request Payload

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

Table 156~~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 ~~creator~~owner 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 157~~156~~: Archive Request Payload

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

Table 158~~157~~: Archive Response Payload

4.22 Recover

This request is used to obtain access to a Managed Object that has been archived. This request MAY require asynchronous polling to obtain the response due to delays caused by retrieving the object from the archive. Once the response is received, the object is now on-line, and MAY be obtained (e.g., via a Get operation). Special authentication and authorization SHOULD be enforced to perform this request (see **[KMIP-UG]**).

Copyright © OASIS® 2010. All Rights Reserved.

Object	Encoding
Unique Batch Item ID	Byte String

Table 173~~172~~: Unique Batch Item ID in Batch Item

6.5 Time Stamp

This field is optionally contained in a client request. It is REQUIRED in a server request and response. 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). Note that the time stamp MAY be used by a client that has no real-time clock, but has a countdown timer, to obtain useful "seconds from now" values from all of the Date attributes by performing a subtraction.

Object	Encoding
Time Stamp	Date-Time

Table 174~~173~~: Time Stamp in Message Header

6.6 Authentication

This is used to authenticate the requester. It is an OPTIONAL information item, depending on the type of request being issued and on server policies. Servers MAY require authentication on no requests, a subset of the requests, or all requests, depending on policy. Query operations used to interrogate server features and functions SHOULD NOT require authentication. The Authentication structure SHALL contain a Credential structure.

The authentication mechanisms are described and discussed in Section 8.

Object	Encoding
Authentication	Structure
Credential	Structure, see 2.1.2, MAY be repeated

Table 175~~174~~: Authentication Structure in Message Header

6.7 Asynchronous Indicator

This Boolean flag indicates whether the client is able to accept an asynchronous response. It SHALL have the Boolean value True if the client is able to handle asynchronous responses, and the value False otherwise. If not present in a request, then False is assumed. If a client indicates that it is not able to handle asynchronous responses (i.e., flag is set to False), and the server is not able to process the request synchronously, then the server SHALL respond to the request with a failure.

Object	Encoding
Asynchronous Indicator	Boolean

Table 176~~175~~: Asynchronous Indicator in Message Request Header

6.8 Asynchronous Correlation Value

This is returned in the immediate response to an operation that is pending and that requires asynchronous polling. Note: the server decides which operations are performed synchronously or asynchronously. A server-generated correlation value SHALL be specified in any subsequent Poll or Cancel operations that pertain to the original operation.

Copyright © OASIS® 2010. All Rights Reserved.