Krishna Yellepeddy

12 August 2010

Draft proposal for Group as a new managed object in KMIP

Use cases for group as a new managed object in KMIP

1. Allow creation of groups of heterogeneous or homogeneous managed objects.
   - Example: Create a homogeneous group of symmetric keys. This set of symmetric keys is treated as a resource and access control is enforced on it. E.g. a set of tape drives of a particular type have access to this group of keys.
   - Create a heterogeneous group of cryptographic objects consisting of asymmetric keys, certificates and secret data. This heterogeneous group may represent a user's credentials for logging on to different applications such as their Gmail account, Facebook account, relational database etc. The user/owner retrieves this group and uses the credentials in the group for signing on to different applications.

2. Assign properties to the group governing how elements in the group are served out for use. This could be thought of as a cursor pattern which is specified at the time of creation of the group. For example:
   a) for a group of symmetric keys, security policy for FIPS compliance may dictate that a key should be served out only once for use by a client for encrypting data. When all the keys have been served out from a group, server returns an error that there are no new keys available. Note that a key can be used any number of times to decrypt data
   b) for a different group of symmetric keys, keys may be served out in a round robin fashion. In this example a key may be served out more than once for encrypting data

3. For a heterogeneous group of elements (e.g. credentials) being managed for a user, user may cache this group of objects and want to be notified by the server if any of the elements in the group have been modified. If they have been modified, then the user refreshes the cache.

Definition of a Group

- A group contains zero or more managed objects, excluding other groups.
- Objects in a group may be heterogeneous or homogeneous. When a group is created, we define whether the group will have heterogeneous or homogeneous objects.
- Group members are represented by uuids for objects that exist on the KMIP server where the group is being created
- Should an object be allowed to belong to more than one group ?
  - It complicates access control
  - Requires complex rules for when an object can belong to two groups. E.g. if group A uses a cursor pattern of using a key only once, but group B uses a round-robin cursor pattern, an object should not be allowed to belong to both groups.
- Recommending that objects should not be allowed to belong to multiple groups.

New attributes needed for Group Object

Attribute Name	Optional/Required	Description
Group Type	Required	Group type is Heterogeneous or Homogeneous. Specified at time of Group creation and cannot be modified after the group is created.
Homogeneity Criteria	Required only if group type is homogeneous.	Specify the list of KMIP attributes with values that must match for homogeneity.
Number of Members	Required	Number of members in the group
Cursor pattern	Defaults to round robin.	Can be “Round robin” or “One time use”. Cannot be modified after the group has been created.
NextMember	Required	UUID of next member that will be served from group, based on the cursor pattern. If the group is empty or there are no more members available to be served, this field is set to null. The usage guide should be updated to state the client SHOULD not use this attribute to get uuid of next member. Instead, they should use the getNextMember operation.
FirstMember	Required	If group is empty, this field is null. Each member in the group has a link to the next member in the group. This link is used to traverse the list of members.
LastMember	Required	If group is empty, this field is null.

Additional attributes for managed objects that are members of a group

- Link: Link is used as a pointer to next member in group and is a new link type. It is modifiable only by the server and not by a client. If this member is the last member, the link attribute for this member is set to null.
- GroupObjectId: UUID of group to which this managed object belongs

Homogeneity criteria

- The criteria for homogeneity is defined at the time a group is created, and cannot be changed after that.
- Homogeneity criteria is defined by specifying those KMIP attribute values that a managed object should match on to be a member of a homogeneous group.
- The next 3 pages lists KMIP attributes that may be used for defining homogeneity.

Attributes that MAY be used for defining criteria for homogeneity

Attribute Name	Used for defining homogeneity criteria	Comments
Unique identifier	N	
Name	N	
Object Type	Y	
Cryptographic algorithm	Y	
Cryptographic length	Y	This attribute should go hand in hand with cryptographic algorithm, otherwise it is not useful as a homogeneity criterion
Cryptographic Parameters	Y	This attribute should go hand in hand with cryptographic algorithm, otherwise it is not useful as a homogeneity criterion. Server should enforce that if cryptographic parameters is a homogeneity criterion, cryptographic algorithm should also be a homogeneity criterion.
Cryptographic Domain Parameters	N	Since this attribute contains a set of OPTIONAL fields that MAY need to be specified in the Create Key Pair Request Payload, it is not useful as a homogeneity criterion.
Certificate Type	Y	
Certificate Identifier	N	While the issuer DN will be the same for multiple certificates, the serial number of certificates will differ. So this is not useful for defining homogeneity.
Certificate Subject	Y	It could be used, but of limited value.

Attributes that MAY be used for defining criteria for homogeneity (continued)

Attribute Name	Used for defining homogeneity criteria	Comments
Certificate Issuer	Y	
Digest	N	
Operation Policy Name	Y	
Cryptographic Usage Mask	Y	
Lease Time	N	
Usage Limits	N	
State	N	
Initial Date	N	
Activation Date	N	
Process Start Date	N	
Protect Stop Date	N	
Deactivation Date	N	
Destroy Date	N	
Compromise Occurrence Date	N	

Attributes that MAY be used for defining criteria for homogeneity (continued)

Attribute Name	Used for defining homogeneity criteria	Comments
Compromise Date	N	
Revocation Reason	N	
Archive Date	N	
Object Group	Y	
Link	N	
Application Specific Information	Y	
Contact Information	Y	
Last Change Date	N	
Custom Attribute	Y	

BACKUP

KMIP Operations permitted on a Group object

KMIP Operation	Supported for Group ?	Comments
Create	N/A	
AddToGroup	New operation for Group objects	Add member to a group by specifying the uuid of the member.
RemoveFromGroup	New operation for Group objects	Remove member from a group by specifying the uuid of the member
Create Key Pair	N/A	
Register	Y. Extended to support Groups	Register a group. It has no members at this point. During registration specify whether Group will have heterogeneous objects or homogeneous objects and what the cursor pattern to use is..Use AddToGroup to add members to a group
Re-key	N/A	
Derive Key	N/A	
Certify	N/A	
Locate	Y	
Check	N/A	Check should be against individual members of a group, and only if it is meaningful
Get	Y	Returns the Group managed object including the number of members and the uuid of the first member. It does not perform 'get' of the members. To get a list of uuids of members, use GetMembersInGroup.

KMIP Operations permitted on Group object continued…

KMIP Operation	Supported for Group ?	Comments
GetNextMember	Y	Returns the next member to be served from the group. Updates "NextMember" group attribute. If there are no more elements that can be served from the group, NextMember is set to null.
GetMembersInGroup	Y	GetMembersInGroup takes two optional fields: a) one for specifying a limit on the number of entries to return. b) another for the starting point from which to iterate The uuid of the last entry returned in the list by the server can be used to iterate through remaining entries in the group. The server defines what the upper limit on the number of entries that can be returned. Note that the group is not locked – while a user is iterating through the member list, additional elements may be added to the group.
Get Attributes	Y	Return the attributes for the group, not for individual members of the group
Get Attribute List	Y	Returns attribute list for the group, not for individual members of the group
Add Attribute	Y	Add attribute to the group object, not for individual members of the group
Modify Attribute	Y	Modify attribute for the group, not for individual members of the group
Delete Attribute	Y	Delete attribute for the group, not for individual members of the group
Obtain Lease	N/A	

KMIP Operations permitted on Group object continued…

KMIP Operation	Supported for Group ?	Comments
Get Usage Allocation	N/A	
Activate	Y	Activate does not apply to Templates. So what happens if a group containing just Templates ? One option is to say that activate is a no-op for template members of a group.
Revoke	N/A	
Destroy	Y	Destroy on a group does not destroy members of the group, just the group object; the server would need to remove all links from member objects to the destroyed group. This is an asynchronous call
Archive	Y	All members of the group are archived. If a client wants to archive a single member of a group, they still have the option to do so. Each member's archive flag is set. This is an asynchronous call.
Recover	Y	All members of the group are recovered. If a client wants to recover a single object as opposed to the entire group, they still have the option to do so. This is an Asynchronous call.
Validate	N/A	
Query	Y	
Cancel	N/A	
Poll	N/A	
Notify	Y	
Put	Y	