Draft proposal for supporting groups in KMIP

Krishna Yellepeddy, Tim Hudson, Bruce Rich, Gordon Arnold, Mathias Bjoerkqvist, Robert Haas

Assumptions

- Limit specification changes to a minimum for 1.1 of the specification. No new OBJECT_TYPE or OPERATION. Introduce a minimal set of new attributes related to groups.
- Allow groups to be heterogeneous and objects to belong to multiple groups, but keep the focus on what is reasonable to accomplish in 1.1.

Assumptions (continued)

- A server implementation may restrict group membership to specific types of keys based on server policy (e.g. to a homogeneous key group) or to a single group, and may have rules for modifications or deletions to members of a group that are outside the scope of the specification.

Use cases

1. Server returns objects from a group that have already been created by the server. Clients rely on the server to generate the right kind of object that they need.
2. Server returns objects from a group that have been registered at the server
3. Server generates objects on the fly when a client specifies a flag to make a request for objects that do not exist in a group.

Use cases (continued)

- Operations on groups
  1. Return ‘fresh’ object from group. An object is ‘fresh’ if it has not been retrieved by a client with a Get call.
  2. Return ‘default’ object from group. ‘default’ is defined by server policy.

Group identification

- Group Identification
  - A simple attribute of type text string
    - We propose the use of existing ‘object Group’ attribute for this. Note that an object can belong to multiple groups.
- Group Member Identification
  - Has two values - ‘fresh’, ‘default’
    - A simple attribute of type enumeration used by a client to indicate whether it wants a fresh/default key

Client call to create a group

- To “create” a group
  - During REGISTER or CREATE of an object, specify Object Group
    - Server policy handles how the group is established and what ‘rules’ it operates under. The policy is outside the scope of the KMIP specification.
    - “DEFAULT” is a reserved object group name.

Client call to add members to group or modify a member

- To add a new object to a group
  - REGISTER (or CREATE)
    - Including the usual options
    - Specify existing Object Group
  - ADD_ATTRIBUTE or MODIFY_ATTRIBUTE
    - Specify existing Object Group

Server policy handles how the group is established and what 'rules' it operates under – e.g. Can you add a member to an existing group if it is not of the same 'type' as the existing members ? Can you remove a member from a group ? These are outside the scope of the KMIP specification.

Fresh and Default group member

- Group Member Identification
  - Add attribute “Object Group Member” of type enumeration
    - GROUP_MEMBER_FRESH
    - GROUP_MEMBER_DEFAULT

Locate ‘fresh’ key

- To get ‘fresh’ key
  - LOCATE
    - Specify the usual options for Locate which can indicate the desired object type. E.g., If you want an AES-128 key you can ask for that explicitly, or not specify it and get whatever the server returns
    - Specify Object Group
    - Specify Object Group Member = ‘fresh’
    - If there are no more fresh keys in a group, server may choose to generate a new key on the fly or not, based on server policy.

Locate ‘default’ key

- To get ‘default’ key
  - LOCATE
    - Specify the usual options which can indicate the desired object type. E.g., If you want an AES-128 key you can ask for that explicitly or not specify it and you’ll get whatever the server returns.
    - Specify Object Group
    - Specify Object Group Member = ‘default’
    - Optionally use “Maximum Items” in the LOCATE if the client only wants a single object to be returned
    - Server locates the ‘default’ key as defined by server policy.

New attributes for managed objects

- Introducing two new optional attributes:
  - Fresh: boolean valued attribute. Set to true to indicate this key has not been used, false otherwise. This attribute is used by the server to keep track of objects in a group that are ‘fresh’.
    - Note that when a client Registers an object without key material, the server does not keep track of whether this object is ‘fresh’.
  - Replicated: integer valued attribute. Set by the server to track the number of different key servers this has been replicated to.

Interoperability profile for groups

- Operations related to groups are dictated by server policy that is outside the scope of the specification.
  - Consequently, we will not specify an interoperability profile for groups.
  - Instead, we'll provide examples of use cases for groups in the use case document.

Upgrade from economy class to first class

- In a future release, if we add Group as a managed object we can bridge from v1.1 as follows:
  - A future KMIP server with support for group as first class object will continue to support the Locate ‘fresh’ and ‘default’ operations to be backward compatible with a v 1.1 client.
  - Existing groups from a v 1.1 server are migrated into a first class group object.