Term	Definition
Key derivation (derivation)	A function in the lifecycle of keying material; the process by which one or more keys are derived from: 1) Either a shared secret from a key agreement computation or a pre-shared cryptographic key, and 2) Other information.
Key management	The activities involving the handling of cryptographic keys and other related security parameters (e.g., IVs and passwords) during the entire life cycle of the keys, including their generation, storage, establishment, entry and output, and destruction.
Key Management Domain	An instance of a key management system where uniqueness of objects can reasonably be expected. A key management system may comprise multiple logical partitions (Key management Domains) where uniqueness is preserved within each partition but is not required across all partitions.
Key wrapping (wrapping)	A method of encrypting and/or MACing/signing keys.
Message Authentication Code (MAC)	A cryptographic checksum on data that uses a symmetric key to detect both accidental and intentional modifications of data.
PGP Key	A RFC 4880-compliant container of cryptographic keys and associated metadata. Usually text-based (in PGP-parlance, ASCII-armored).
Private key	A cryptographic key used with a public key cryptographic algorithm that is uniquely associated with an entity and is not made public. The private key is associated with a public key. Depending on the algorithm, the private key MAY be used to: 1. Compute the corresponding public key, 2. Compute a digital signature that can be verified by the corresponding public key, 3. Decrypt data that was encrypted by the corresponding public key, or 4. Compute a piece of common shared data, together with other information.
Profile	A specification of objects, attributes, operations, message elements and authentication methods to be used in specific contexts of key management server and client interactions (see **[KMIP-Prof]**).
Public key	A cryptographic key used with a public key cryptographic algorithm that is uniquely associated with an entity and that MAY be made public. The public key is associated with a private key. The public key MAY be known by anyone and, depending on the algorithm, MAY be used to: 1. Verify a digital signature that is signed by the corresponding private key, 2. Encrypt data that can be decrypted by the corresponding private key, or 3. Compute a piece of shared data.
Public key certificate (certificate)	A set of data that uniquely identifies an entity, contains the entity's public key and possibly other information, and is digitally signed by a trusted party, thereby binding the public key to the entity.

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

Table 30: Activation Date Attribute Rules

4.2 Alternative Name

The *Alternative Name* attribute is used to identify and locate the object. This attribute is assigned by the client, and the *Alternative Name Value* is intended to be in a form that humans are able to interpret. The key management system MAY specify rules by which the client creates valid alternative names. Clients are informed of such rules by a mechanism that is not specified by this standard. Alternative Names MAY NOT be unique within a given ~~key management domain~~Key Management Domain.

Item	Encoding	REQUIRED
Alternative Name	Structure	
Alternative Name Value	Text String	Yes
Alternative Name Type	Enumeration	Yes

Table 31: Alternative Name Attribute Structure

SHALL always have a value	No
Initially set by	Client
Modifiable by server	Yes (Only if no value present)
Modifiable by client	Yes
Deletable by client	Yes
Multiple instances permitted	Yes
Applies to Object Types	All Objects

Table 32: Alternative Name Attribute Rules

4.3 Always Sensitive

The server SHALL create this attribute, and set it to True if the Sensitive attribute has always been True. The server SHALL set it to False if the Sensitive attribute has ever been set to False.

Item	Encoding
Sensitive	Boolean

Encoding	Description
Text String	Unique Identifier of a Managed Object.
Enumeration	Unique Identifier Enumeration
Integer	Zero based nth Unique Identifier in the response. If negative the count is backwards from the beginning of the current operation's batch item.

Table 89: Linked Object Identifier encoding descriptions

Item	Encoding	REQUIRED
Link	Structure	
Link Type	Enumeration	Yes
Linked Object Identifier	Text String/Enumeration/Integer	Yes

Table 90: 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, Re-key Key Pair, Register
Applies to Object Types	All Objects

Table 91: Link Attribute Structure Rules

4.32 Name

The *Name* attribute is a structure used to identify and locate an object. This attribute is assigned by the client, and the *Name Value* is intended to be in a form that humans are able to interpret. The key management system MAY specify rules by which the client creates valid names. Clients are informed of such rules by a mechanism that is not specified by this standard. Names SHALL be unique within a given ~~key management domain~~Key Management Domain, but are NOT REQUIRED to be globally unique.

Item	Encoding	REQUIRED
Name	Structure	
Name Value	Text String	Yes
Name Type	Enumeration	Yes

Table 92: Name Attribute Structure

SHALL always have a value	No
Initially set by	Server
Modifiable by server	Yes
Modifiable by client	No
Deletable by client	No
Multiple instances permitted	No
When implicitly set	When object is rotated by the server
Applies to Object Types	All Objects

Table 131: Rotate Latest Attribute Rules

4.52 Rotate Name

The *Rotate Name* attribute is used to identify a set of managed objects that have been rotated. This attribute is assigned by the client, and the *Rotate Name Value* is intended to be in a form that humans are able to interpret. The key management system MAY specify rules by which the client creates valid rotate names. Clients are informed of such rules by a mechanism that is not specified by this standard. Rotate Names MAY NOT be unique within a given ~~key management domain~~Key Management Domain.

Item	Encoding	Encoding
Rotate Name	Structure	
Rotate Name Value	Text String	Yes
Rotate Name Type	Enumeration	Yes

Table 132: Rotate Name Attribute Structure

SHALL always have a value	No
Initially set by	Client
Modifiable by server	No
Modifiable by client	Yes
Deletable by client	Yes
Multiple instances permitted	No
Applies to Object Types	All Objects

Table 133: Rotate Name Attribute Rules

4.53 Rotate Offset

When automatic rotation of the Managed Object is performed by the server, specifies the Offset value to use in the equivalent of the ReKey, ReKeyKeyPair or ReCertify operation performed by the server.

Item	Encoding
Rotate Offset	Interval

Table 134: Rotate Offset 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	When object is created or registered
Applies to Object Types	All Objects

Table 135: Rotate Offset Attribute Rules

4.54 Sensitive

If True then the server SHALL prevent the object value being retrieved (via the Get operation) unless it is wrapped by another key. The server SHALL set the value to False if the value is not provided by the client.

Item	Encoding
Sensitive	Boolean

Table 136: Sensitive Attribute

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	When object is created or registered
Applies to Object Types	All Objects

Table 137: Sensitive Attribute Rules

4.55 Short Unique Identifier

The *Short Unique Identifier* is generated by the key management system to uniquely identify a Managed Object using a shorter identifier. It is only REQUIRED to be unique within the identifier space managed by a single key management system, however this identifier SHOULD be globally unique in order to allow for a ~~key management domain~~Key Management Domain export of such objects. This attribute SHALL be assigned by the key management system upon creation or registration of a *Unique Identifier*, and then SHALL NOT be changed or deleted before the object is destroyed.

The *Short Unique Identifier* SHOULD be generated as a SHA-256 hash of the *Unique Identifier* and SHALL be a 32 byte *byte string*.

7. The transition from Deactivated to Destroyed is caused by a client issuing a Destroy operation, or by a server, both in accordance with server policy. The server destroys the object when (and if) server policy dictates.
8. The transition from Deactivated to Compromised is caused by a client issuing a Revoke operation with a Revocation Reason containing a *Revocation Reason Code* of Compromised.
9. The transition from Compromised to Destroyed Compromised is caused by a client issuing a Destroy operation, or by a server, both in accordance with server policy. The server destroys the object when (and if) server policy dictates.
10. The transition from Destroyed to Destroyed Compromised is caused by a client issuing a *R*evoke operation with a Revocation Reason containing a *Revocation Reason Code* of Compromised.

Only the transitions described above are permitted.

Item	Encoding
State	Enumeration

Table 140: State Attribute

SHALL always have a value	Yes
Initially set by	Server
Modifiable by server	Yes
Modifiable by client	No, but only by the server in response to certain requests (see above)
Deletable by client	No
Multiple instances permitted	No
When implicitly set	Create, Create Key Pair, Register, Derive Key, Activate, Revoke, Destroy, Certify, Re-certify, Re-key, Re-key Key Pair
Applies to Object Types	All Objects

Table 141: State Attribute Rules

4.57 Unique Identifier

The *Unique Identifier* is generated by the key management system to uniquely identify a Managed Object. It is only REQUIRED to be unique within the identifier space managed by a single key management system, however this identifier SHOULD be globally unique in order to allow for a ~~key management domain~~Key Management Domain export of such objects. 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.