Local Rights Manager for Secure Content Exchange

Approved Version 1.0 – 05 Jul 2011

Open Mobile Alliance

OMA-TS-SCE_LRM-V1_0-20110705-A

Contents

Figures

Tables

1. Scope

Open Mobile Alliance (OMA) specifications are the result of continuous work to define industry-wide interoperable mechanisms for developing applications and services that are deployed over wireless communication networks.

The scope of OMA Secure Content Exchange (SCE) is to enhance the OMA Digital Rights Management v2.1 [DRM-DRM-v2.1] specifications to enable the secure exchange of DRM Content among multiple devices. These enhancements include the following:

- New capabilities that enable flexible sharing of purchased content in ways that were not possible using Domains as defined in OMA DRM v2.1. These new features include Copy and Move of Rights between Devices, Lending and Sharing in an ad hoc manner.
- Extension of the OMA DRM v2.1 Domain concept to the User Domain concept, which allows different Rights Issuers to generate Rights Objects for the same User Domain.
- The definition of the Import function allows content protected by Non-OMA DRM mechanisms to be consumed by SCE Devices. Together with the Export function from OMA DRM v2.1, the Import function allows OMA SCE Devices to securely exchange content with Non-OMA DRM devices.
- Enhancements to the OMA DRM specifications to enable consumption of DRM Content contained in an MPEG-2 Transport Stream across a wide variety of user Devices.

This document specifies the Local Rights Manager (LRM), which is used for the conversion of Non-OMA DRM protected content to OMA DRM protected content.

2. References

2.1 Normative References

[AES-WRAP] Advanced Encryption Standard (AES) Key Wrap Algorithm. RFC 3394, J. Schaad and R. Housley, September 2002, URL:http://tools.ietf.org/html/rfc3394

[CERT-PROF] "Certificate and CRL Profiles", OMA-Security-CertProf-v1_1, Open Mobile Alliance, URL:http://www.openmobilealliance.org/

[DRM-v2.1] The OMA DRM 2.1 enabler as described in "Enabler Release Definition for DRM V2.1, Approved Version 2.1", OMA-TS-DRM-DRM-V2_0-20060303-A, Open Mobile Alliance™, URL:http://www.openmobilealliance.org/

[DRM-DCF-v2.1] "DRM Content Format, Approved Version 2.1", OMA-TS-DRM-DCF-V2_0-20060303-A, Open Mobile Alliance™, URL:http://www.openmobilealliance.org/

[DRM-DRM-v2.0] "DRM Specification", Open Mobile Alliance™, OMA-TS-DRM-DRM-V2_0-20060303-A, URL:http://www.openmobilealliance.org/

[DRM-DRM-v2.1] "DRM Specification, Approved Version 2.1", OMA-TS-DRM-DRM-V2_0-20060303-A, Open Mobile Alliance™, URL:http://www.openmobilealliance.org/

[DRM-REL-v2.1] "DRM Rights Expression Language, Approved Version 2.1", OMA-TS-DRM-REL-V2_0-20060303-A, Open Mobile Alliance™, URL:http://www.openmobilealliance.org/

[RFC2104] "HMAC: Keyed-Hashing for Message Authentication", H. Krawczyk, M. Bellare, and R. Canetti, February 1997, URL:http://tools.ietf.org/html/rfc2104

[RFC2119] "Key words for use in RFCs to Indicate Requirement Levels", S. Bradner, March 1997, URL:http://tools.ietf.org/html/rfc2119

[RFC2234] "Augmented BNF for Syntax Specifications: ABNF". D. Crocker, Ed., P. Overell. November 1997, URL:http://www.ietf.org/rfc/rfc2234.txt

[RFC3280] Housley, R., Polk, W., Ford, W. and D. Solo, "Internet Public Key Infrastructure - Certificate and Certificate Revocation List (CRL) Profile", April 2002. URL:http://www.ietf.org/rfc/rfc3280.txt

[RFC3447] "Public-Key Cryptography Standards (PKCS) #1: RSA Cryptography Specifications Version 2.1", J. Jonsson, B. Kaliski, February 2003, URL:http://tools.ietf.org/html/rfc3447

[SCE-DOM] "SCE User Domains", OMA-TS-SCE-DOM-Vx_y-D, Open Mobile Alliance™, URL:http://www.openmobilealliance.org/

[SCE-GEN] "SCE Generic Mechanisms, Draft Version", OMA-TS-SCE_GEN-Vx_y-D, Open Mobile Alliance™, URL:http://www.openmobilealliance.org/

[SCE-REL] "DRM Rights Expression Language – SCE Extensions", OMA-TS-SCE-REL-Vx_y-D, Open Mobile Alliance™, URL:http://www.openmobilealliance.org/

[SCR-RULES] "SCR Rules and Procedures", Open Mobile Alliance™, OMA-ORG-SCR_Rules_and_Procedures, URL:http://www.openmobilealliance.org/

[SHA1] NIST FIPS 180-2: Secure Hash Standard. August 2002, URL:http://csrc.nist.gov/publications/fips/fips180-2/fips180-2withchangenotice.pdf

[XC14N] Exclusive XML Canonicalization: Version 1.0, John Boyer, Donald E. Eastlake 3rd and Joseph Reagle, W3C Recommendation 18 July 2002, URL:http://www.w3.org/TR/xml-exc-c14n

2.2 Informative References

[OMA-DICT] "Dictionary for OMA Specifications", Version x.y, Open Mobile Alliance™, OMA-ORG-Dictionary-Vx_y, URL:http://www.openmobilealliance.org

[SCE-AD] "Secure Content Exchange Architecture, Draft Version", OMA-AD-SCE-Vx_y-D, Open Mobile Alliance™, URL:http://www.openmobilealliance.org/

3. Terminology and Conventions

3.1 Conventions

The key words “MUST”, “MUST NOT”, “REQUIRED”, “SHALL”, “SHALL NOT”, “SHOULD”, “SHOULD NOT”, “RECOMMENDED”, “MAY”, and “OPTIONAL” in this document are to be interpreted as described in [RFC2119].

All sections and appendixes, except “Scope” and “Introduction”, are normative, unless they are explicitly indicated to be informative.

3.2 Definitions

Composite Object	A content object that contains one or more Media Objects by means of inclusion.
Constraint	A restriction on the Permission over DRM Content (DRM V2.0).
Consume	To Play, Display, Print or Execute DRM Content on a Device or to render DRM Content on a Render Client.
Copy	To make Rights existing on a source Device available for use by a recipient Device, without affecting availability on the source Device. Rights may be restricted on the recipient Device. Note: this is different from the V2.1 definition.
Device	A Device is the entity (hardware/software or combination thereof) within a user equipment that implements a DRM Agent. The Device is also conformant to the OMA DRM specifications. The Device may include a smart card module (e.g. a SIM) (DRM V2.0).
Device Rights Object	A Rights Object that is initially targeted to a specific entity. Subsequently, the Rights Object may be allowed to be targeted to other entities to be consumed, serially or in parallel, independently of membership in a Domain or User Domain.
Domain	A set of v2.x and/or SCE DRM Agents that can consume Domain Rights Objects.
Domain Authority	The entity to specify the Domain Policy for a User Domain or an Ad Hoc Domain.
Domain Enforcement Agent	The entity to enforce the Domain Policy on behalf of the Domain Authority. It may reside in the network as a service or in a User’s device.
Domain Policy	A collection of attributes which defines the policy determining characteristics of the membership of a User Domain or Ad Hoc Domain, as set by the Domain Authority that the Domain Enforcement Agent will enforce.
Domain Rights Object	A Rights Object that is targeted to a specific v2.x Domain. The Rights Object can be consumed independently by each v2.x or SCE DRM Agent that is a member of the Domain.
DRM Agent	The entity in the Device that manages Permissions for Media Objects on the Device (DRM V2.1). In this document, the DRM Agent implements some or all the functionality defined in this specification.
DRM Content	Media Objects that are consumed according to a set of Permissions in a Rights Object (DRM V2.0).
DRM Time	A secure, non user-changeable time source. The DRM Time is measured in the UTC time scale (DRM V2.0).
Execute	To execute a software programme (DRM V2.0).
Import	To convert Import-Ready Data into OMA (P)DCFs and ROs.
Import-Ready Data	Content and associated Rights derived from Non-OMA DRM-sourced data that can be converted into OMA (P)DCFs and ROs.
Imported-Content	OMA (P)DCF(s) resulting from converting Import-Ready Data.
Imported-Data	Imported-Content and associated Imported-Rights-Object(s).
Imported-Rights-Object	An OMA RO resulting from converting Import-Ready Data.
Lending	The act of sharing such that the Shared Rights cannot be used on the source Device as long as the recipient Device is able to render the shared Content associated with the Shared Rights.

Local Rights Manager (LRM)	An entity that is responsible for aspect(s) of Import and it may also manage an Imported-Content for a limited group of OMA DRM Agents.
Media Object	A digital work e.g. a ringing tone, a screen saver, a Java game or a Composite Object (DRM V2.1)
Move	To make Rights existing initially on a source Device fully or partially available for use by a recipient Device, such that the Rights or parts thereof that become usable on the recipient Device can no longer be used on the source Device.
Non-OMA DRM	A protection system other than OMA DRM, which may include copy protection mechanisms for storage medium and/or transport mechanisms.
Permission	Actual usage or activities allowed (by a Rights Issuer or Local Rights Manager) over DRM Content.
Play	To create a transient, perceivable rendition of a resource.
Print	To create a fixed and directly perceivable rendition of a resource.
Render Client	The entity (hardware, software or combination thereof) within a user equipment that implements a Render Agent. The Render Client is used to transiently render DRM Content.
Rights	The collection of permissions and constraints defining under which circumstances access is granted to DRM Content.
Rights Object	A collection of Permissions and other attributes which are linked to DRM Content.
Set-top Box	A device capable of receiving digital broadcast services contained in an MPEG-2 transport stream that may be delivered over cable, satellite, terrestrial, IP or any other medium. To access the digital broadcast services, a Set-top Box (STB) may or may not use a Conditional Access System. A STB may or may not be OMA DRM compliant.
Shared Rights	Rights that can be consumed on multiple Devices, where the allowed distribution and consumption of the Rights among the Devices are specified by permissions in the Rights themselves or in the Domain Policy of the Domain for which the Rights were obtained.
Sharing	The act of providing Shared Rights from a source Device to a recipient Device, such that the recipient Device is able to render the shared content associated with the Shared Rights.
User	The human user of a Device. The User does not necessarily own the Device (DRM V2.0).
User Domain	A set of v2.x and/or SCE DRM Agents that can consume User Domain Rights Objects.
User Domain Rights Object	A Rights Object that is targeted to a specific User Domain. Besides requiring membership in the User Domain, consumption may require being targeted to an SCE DRM Agent.

3.3 Abbreviations

AES	Advanced Encryption Standard
CA	Certification Authority
CEK	Content Encryption Key
CRL	Certificate Revocation List
DCF	DRM Content Format
DEA	Domain Enforcement Agent
DER	Distinguished Encoding Rules
DRM	Digital Rights Management
FQDN	Fully Qualified Domain Name
HMAC	Keyed-Hash Message Authentication Code
KDF	Key Derivation Function
LRM	Local Rights Manager
LRMID	LRM Identifier
MAC	Message Authentication Code

MDCF	MPEG-2 Transport Stream DRM Content Format
N/A	Not Applicable
OCSP	Online Certificate Status Protocol
OMA	Open Mobile Alliance
PDCF	Packetized DRM Content Format
(P)DCF	A DCF or a PDCF
PKCS	Public-Key Cryptography Standards
REK	Rights Object Encryption Key
REL	Rights Expression Language
RFC	Request for Comments
RI	Rights Issuer
RO	Rights Object
ROAP	Rights Object Acquisition Protocol
ROID	Rights Object Identifier
SCE	Secure Content Exchange
SCR	Static Conformance Requirement
URI	Uniform Resource Indicator
URL	Uniform Resource Locator
XML	Extensible Markup Language

4. Introduction

The Secure Content Exchange enabler extends the mechanisms described in Digital Rights Management v2.1 [DRM-DRM-v2.1] to increase the flexibility of DRM Content usage. Some of these extensions allow the Import of Non-OMA DRM Content, the exchange of DRM Content among Devices and the implementation of a central domain management function.

This document specifies the Local Rights Manager (LRM), which is used for the conversion of Non-OMA DRM protected content to OMA DRM protected content. The LRM is responsible for creating Imported ROs and converting the content to DCF, PDCF or MDCF.

The LRM can have interfaces to DRM Agents, the Domain Enforcement Agent (DEA) and Rights Issuers (RIs). This document describes the LRM-RI interface (LRM-4-LRMP) and the LRM-DRM Agent interface (SCE-6-LRMP). The LRM-DEA interface (SCE-5-LRMP) is specified in [SCE-DOM].

Depending on the LRM key purpose, the LRM may generate ROs for DRM Agents that are compliant to DRM v2.0 or v2.1 only, i.e. not to SCE. For the case that an LRM does not have a valid key purpose for directly Importing to DRM v2.0 or v2.1 Devices, this specification provides protocols where the LRM can still provide these Devices with Imported-Content, with the assistance of an RI.

The LRM has been designed such that it is very similar to an RI. However, since the LRM can be situated locally (i.e. as a Set-top Box at the user's home), restrictions are provided to limit the danger of a potential compromise.

5. Overview of LRM Functions

Each LRM (as well as each DRM Agent) is required to support DRM Time.

5.1 Creation of Imported-ROs

5.1.1 LRM key purposes and generation of Imported-ROs

The LRM MAY have a combination of three different key purposes: the oma-kp-localRightsManagerDomain key purpose, the oma-kp-localRightsManagerDevice key purpose and the oma-kp-rightsIssuer key purpose. However, the LRM MUST have at least the oma-kp-localRightsManagerDomain key purpose or the oma-kp-localRightsManagerDevice key purpose. This is important so that an SCE Device is able to distinguish an LRM with an oma-kp-rightsIssuer key purpose from an RI.

To generate a Device RO for an SCE Device, the LRM MUST have at least the oma-kp-localRightsManagerDevice key purpose.

To generate a User Domain RO, the LRM MUST have at least the oma-kp-localRightsManagerDomain key purpose.

An LRM MUST NOT Import OMA DRM v2.x Domain ROs to an SCE Device.

An LRM MAY have an oma-kp-rightsIssuer key purpose in order to allow the Import of Device ROs, Domain ROs and User Domain ROs (only non-<userDomain>-constrained ROs) to OMA DRM v2.x Devices. An OMA DRM v2.x Device cannot distinguish an LRM with at least the oma-kp-rightsIssuer key purpose from an RI. The OMA DRM v2.x Device will therefore accept these ROs independently of the other key purposes of the LRM. However, SCE Devices MUST take all the key purposes into account.

If an LRM key purpose (oma-kp-localRightsManagerDevice or oma-kp-localRightsManagerDomain) is present, an SCE Device disregards an oma-kp-rightsIssuer key purpose if present.

6. LRM and DEA Protocols

6.1 SCE-4-LRMP

This section defines the protocols by which an LRM communicates with an RI. The protocols include LRM-RI Registration protocol, LRM-RI Create Device RO protocol and so on.

6.1.1 LRM-RI Registration Protocol

The LRM-RI Registration protocol is a complete security information exchange and handshake between the RI and the LRM and is generally only executed at first contact, but may also be executed when there is a need to update the exchanged security information, or when DRM Time in the LRM is deemed inaccurate by the Rights Issuer. This protocol includes negotiation of protocol parameters and protocol version, cryptographic algorithms, exchange of certificate preferences, optional exchange of certificates, mutual authentication of LRM and RI, and integrity protection of protocol messages.

Successful completion of the Registration protocol results in the establishment of an RI Context in the LRM containing RI-specific security related information such as agreed protocol parameters, protocol version, and certificate preferences. An RI Context is necessary for execution of the other protocols in the SCE-4-LRMP suite. Figure 1 depicts the 4-pass LRM-RI Registration protocol.

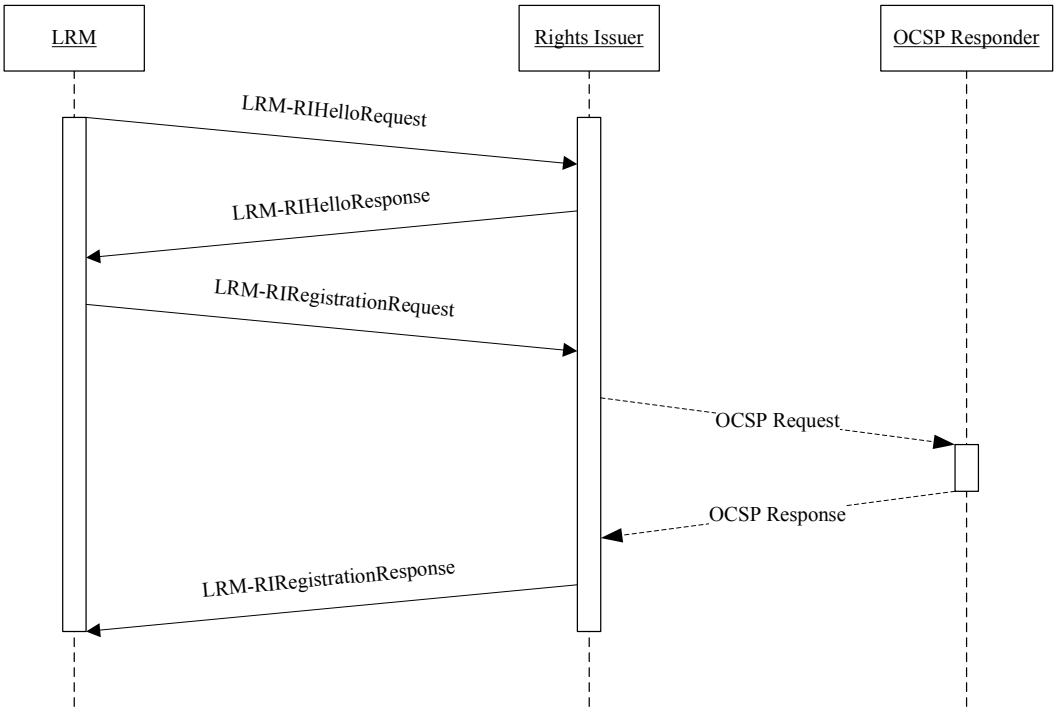

Figure 1 – The 4-pass LRM-RI Registration Protocol

6.1.1.1 Trigger for LRM-RI Registration Protocol

An LRMRIRegistrationTrigger MAY be delivered from an RI to an LRM to invoke the LRM-RI registration protocol. The root element of the message MUST be a <gen:drmTrigger> element as specified in [SCE-GEN], assigning the fixed value "**LRMRIRegistration**" for the **type** attribute.

The <gen:trgInfo> element under the <gen:body> element MAY have an <lrm:LRMRIRegistrationTriggerInformation> child element.

```
<element name="LRMRIRegistrationTriggerInformation">
   <complexType>
     <sequence>
       <element name="LRMID" type="ID" minOccurs="0"/>
     </sequence>
   </complexType>
</element>
```

The <LRMID> element contains the LRM's Identifier.

Appendix D.1 shows an example of an LRMRIRegistrationTrigger. If an LRM receives an LRMRIRegistrationTrigger, it MUST check if the type attribute has "LRMRIRegistrationTrigger" and if the value of <LRMID> is equal to one of the LRM's ID. If the checking fails, the LRM ignores the trigger. If the trigger message is verified, the LRM MUST invoke the LRMRIRegistration protocol by sending an LRMRIRegistrationRequest message to the Rights Issuer (to the address indicated by <reqURL> element).

Before initiating the LRMRIRegistration protocol the LRM MUST obtain user consent before contacting the RI; however, if the FQDN (Fully Qualified Domain Name) part of the <reqURL> element of the LRMRIRegistrationTrigger corresponds to an entry in the User Consent Whitelist the LRM MAY contact the RI without obtaining explicit user consent. A User Consent Whitelist contains the Fully Qualified Domain Name of authorised RIs and the corresponding Rights Issuer's identifier. LRM SHOULD implement a User Consent Whitelist.

6.1.1.2 LRM-RI Hello Request

The LRM-RI Hello Request message is sent from the LRM to the Rights Issuer to initiate the 4-pass LRM-RI Registration protocol. This message expresses LRM information and preferences. The request message is an element of type gen:Request, in which the elements are the same as specified in [SCE-GEN].

6.1.1.3 LRM-RI Hello Response

The LRM-RI Hello Response message is the second message of the 4-pass LRM-RI Registration protocol and is sent from the Rights Issuer to the LRM in response to an LRM-RI Hello Request message. The message expresses RI preferences and decisions based on the values supplied by the LRM. The response message is an element of type gen:Response, in which the elements are the same as specified in [SCE-GEN].

6.1.1.4 LRM-RI Registration Request

An LRM sends the LRM-RI Registration Request message to an RI to request registration with the RI. The message is sent as the third message in the 4-pass LRM-RI Registration protocol. The root element of the message MUST be an <LRMRIRegistrationRequest> element of type gen:Request, in which the following elements are present:

Table 1: LRM-RIRegistrationRequest Message Parameters

element / attribute	usage	value
sessionID	M	Default, as specified in [SCE-GEN]
reqID	M	Default, as specified in [SCE-GEN]
resID	M	Default, as specified in [SCE-GEN]
nonce	M	Default, as specified in [SCE-GEN]
time	M	Default, as specified in [SCE-GEN]
certificateChain	O	Default, as specified in [SCE-GEN]
reqInfo	M	Specified below
triggerNonce	O	Default, as specified in [SCE-GEN]
signature	M	Default, as specified in [SCE-GEN]

The <gen:reqInfo> element under the <LRMRIRegistrationRequest> element MUST contain an <lrm:LRMRIRegistrationRequestInformation> child element as defined by the following XML schema fragment:

```
<element name="LRMRIRegistrationRequest" type="gen:Request"/>
 <element name="LRMRIRegistrationRequestInformation">
  <complexType>
   <sequence>
    <element name=" supportedUpstreamDRMs" type="lrm:SetOfDRMSystem"/>
    <element name=" needMoveService"  minOccurs="0"/>
   </sequence>
  </complexType>
 </element>

<complexType name="SetOfDRMSystem">
<sequence>
<element name="supportedDRMSystem" type="string" maxOccurs="unbounded"/>
    </sequence>
</complexType>
```

Besides the elements included in gen:RegReqInfo, the <reqInfo> element in LRM-RIRegistrationRequest message includes further <supportedUpstreamDRMs> element and one optional <needMoveService> element.

The <supportedUpstreamDRMs> identifies the upstream DRM systems that are supported by the LRM, i.e. the LRM can Import ROs from these DRM systems.

The <needMoveService> element, if present, is used by the LRM to indicate to the RI that the LRM needs the 'RI provides Move' service for the ROs created by the LRM, so that the ROs created by the LRM can be Moved via the RI to other Devices.

6.1.1.5 LRM-RI Registration Response

The LRM-RI Registration Response message is sent from the Rights Issuer to the LRM in response to an LRM-RI Registration Request message. This message completes the Registration protocol, and if successful, enables the LRM to establish an RI Context for this RI. The root element of the message MUST be an <LRMRIRegistrationResponse> element of type gen:Response, in which the following elements are present:

Table 2: LRM-RIRegistrationResponse Message Parameters

element / attribute	usage	value
status	M	Default, as specified in [SCE-GEN]
sessionID	M	Default, as specified in [SCE-GEN]
errorMessage	O	Default, as specified in [SCE-GEN]
errorRedirectURL	O	Default, as specified in [SCE-GEN]
reqID	M	Default, as specified in [SCE-GEN]
resID	M	Default, as specified in [SCE-GEN]
certificateChain	O	Default, as specified in [SCE-GEN]
ocspResponse	O	Specified below
rspInfo	M	Specified below
signature	M	Specified below

ocspResponse MUST be present when the RI deems that the LRM's DRM Time is inaccurate. If the LRM receives this parameter, it MUST adjust its DRM Time for the current trust model to the time in the producedAt component of the nonce-based OCSP response. The nonce in the OCSP response MUST be equal to the nonce sent in the preceding LRM-RIRegistrationRequest message. If and only if the nonces match, the LRM SHALL use the OCSP response to update its DRM Time. For further information on the <ocspResponse> element, please refer to [SCE-GEN].

The <gen:resInfo> element under the <LRMRIRegistrationResponse> element MUST contain an <lrm:LRMRIRegistrationResponseInformation> child element as defined by the following XML schema fragment:

```
<element name="LRMRIRegistrationResponse" type="gen:Response"/>
 <element name="LRMRIRegistrationResponseInformation">
  <complexType>
   <sequence>
    <element name="selectedUpstreamDRMs" type="lrm:SetOfDRMSystem" minOccurs="0"/>
    <element name="provideMoveService" minOccurs="0"/>
   </sequence>
  </complexType>
 </element>
```

Besides the elements included in gen:ResRegInfo, the <rspInfo> element in LRMRIRegistrationResponse message includes further <selectedUpstreamDRMs> element and an optional <provideMoveService> element.

The <selectedUpstreamDRMs> specifies the upstream DRM systems that will be supported by the RI.

The <provideMoveService> element is used by the RI to indicate to the LRM whether the RI will provide Move service for the ROs that the LRM creates:

- If the <provideMoveService> element is present in rspInfo element in LRM-RIRegistrationResponse, the LRM MAY indicate within all the Imported-Rights-Objects that the LRM creates that this particular Rights Issuer is eligible to Move the Rights.
- If the <provideMoveService> element is NOT present in rspInfo element in LRM-RIRegistrationResponse, the LRM SHALL NOT indicate within any Imported-Rights-Object that the LRM creates that this particular Rights Issuer is eligible to Move the Rights.

6.1.2 LRM-RI DevPubKeyAcquisition Protocol

The 2-pass LRM-RI DevPubKeyAcquisition protocol is the protocol by which an LRM gets the public key of an OMA DRM v2.x Device from the RI. The RI can acquire the Public Key of that OMA DRM v2.x Device through a 4-pass ROAP Registration with that Device.

Successful completion of this protocol results in the establishment of a Device Context in the LRM containing Device-specific information including the public key of that OMA DRM v2.x Device. The Device Context is necessary for execution of the LRM-RI Create Device RO protocol.

This protocol can be initiated by a ROAP Trigger{LRM-RI DevPubKeyAcquisition trigger}, see section 6.1.2.1.

This protocol MAY involve OCSP protocol between Rights Issuer and OCSP Responder for checking status of Rights Issuer's certificate chain.

Figure 2 depicts the 2-pass LRM-RI DevPubKeyAcquisition protocol.

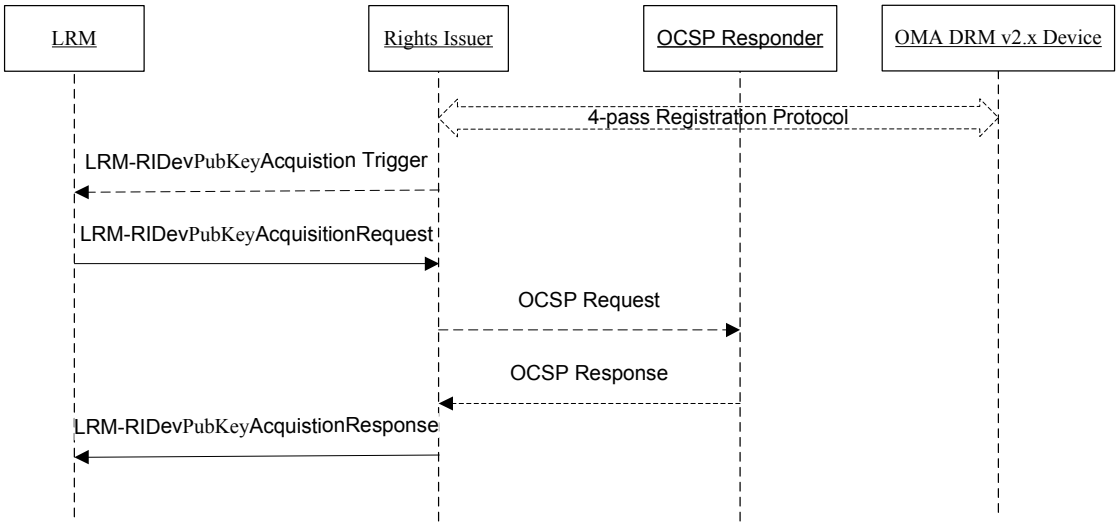

Figure 2 – The 2-pass LRM-RI DevPubKeyAcquisition Protocol

6.1.2.1 Trigger for LRM-RI DevPubKeyAcquisition Protocol

An LRMRIDevPubKeyAcquisitionRegistrationTrigger MAY be delivered from an RI to an LRM to invoke the LRM-RI DevPubKeyAcquisition protocol. The root element of the message MUST be a <gen:drmTrigger> element as specified in [SCE-GEN], assigning the fixed value "**LRMRIDevPubKeyAcquisitionRegistration**" for the **type** attribute.

The <gen:trgInfo> element under the <gen:body> element MUST contain an <lrm:LRMRITriggerInformation> child element.

```
<element name="LRMRIDevPubKeyAcquisitionTriggerInformation">
  <complexType>
    <sequence>
      <element name="DevID" type="ID"/>
    </sequence>
  </complexType>
```

```
</element>
```

The type attribute of the message SHALL be "LRM-RIDevPubKeyAcquisitionTrigger"

The version attribute of the message SHALL be "1.0 ".

The <resID> element MUST contain the Rights Issuer's identifier.

The <reqURL> element MUST contain the Rights Issuer's URL address that serves LRM-RI DevPubKeyAcquisition protocol.

The <DevID> element MUST contain the Identifier of the OMA DRM v2.x Device.

When the LRM receives the ROAP LRM-RI DevPubKeyAcquisitionTrigger, it initiates the ROAP LRM-RI DevPubKeyAcquisition protocol exchange as soon as possible.

6.1.2.2 LRM-RI DevPubKeyAcquisition Request

An LRM sends the LRM-RI DevPubKeyAcquisition Request message to an RI to request the Public Key of an OMA DRM v2.x Device through the RI. The root element of the message MUST be an <LRMRIDevPubKeyAcquisitionRequest> element of type gen:Request, in which the following elements are present:

Table 3: LRM-RI DevPubKeyAcquisition Request Message Parameters

element / attribute	usage	value
triggerNonce	O	Default, as specified in [SCE-GEN]
reqID	M	Default, as specified in [SCE-GEN]
resID	M	Default, as specified in [SCE-GEN]
nonce	M	Default, as specified in [SCE-GEN]
time	M	Default, as specified in [SCE-GEN]
certificateChain	O	Default, as specified in [SCE-GEN]
DevID	M	Default, as specified in [SCE-GEN]
signature	M	Default, as specified in [SCE-GEN]

The <gen:reqInfo> element under the <LRMRIDevPubKeyAcquisitionRequest> element MUST contain an <lrm:LRMRIDevPubKeyAcquisitionRequestInformation> child element as defined by the following XML schema fragment:

```
<element name="LRMRIDevPubKeyAcquisitionRequest" type="gen:Request"/>
  <element name="LRMRIDevPubKeyAcquisitionRequestInformation">
    <complexType>
      <sequence>
        <element name="DevID" type="ID"/>
      </sequence>
    </complexType>
</element>
```

DevID is the identifier of the OMA DRM v2.x Device. If this protocol is initiated by an LRM-RI DevPubKeyAcquisition Trigger, the DevID in this message is identical to that DevID in the LRM-RI DevPubKeyAcquisition Trigger.

6.1.2.1 LRM-RI DevPubKeyAcquisition Response

The LRM-RI DevPubKeyAcquisition Response message is sent from the Rights Issuer to the LRM in response to an LRM-RI DevPubKeyAcquisition Request message. This message completes the LRM-RI DevPubKeyAcquisition protocol, and if successful, enables the LRM to establish a Device Context for that OMA DRM v2.x Device. The root element of the message MUST be an <LRMRIDevPubKeyAcquisitionResponse> element of type gen:Response, in which the following elements are present:

Table 4: LRM-RI DevPubKeyAcquisition Response Message Parameters

element / attribute	usage	value
status	M	Result of the rdpDropDomainRequest processing.
errorMessage	O	Default, as specified in [SCE-GEN]
errorRedirectURL	O	Default, as specified in [SCE-GEN]
reqID	M	Default, as specified in [SCE-GEN]
resID	M	Default, as specified in [SCE-GEN]
DevPubKey	M	string
CertificateChain	O	Default, as specified in [SCE-GEN]
ocspResponse	O	Default, as specified in [SCE-GEN]
signature	M	Specified below

The <gen:resInfo> element under the <LRMRIDevPubKeyAcquisitionResponse> element MUST contain an <lrm:LRMRIDevPubKeyAcquisitionResponseInformation> child element as defined by the following XML schema fragment:

```
<element name="LRMRIDevPubKeyAcquisitionResponse" type="gen:Response"/>
  <element name="LRMRIDevPubKeyAcquisitionResponseInformation">
    <complexType>
      <sequence>
        <element name="DevPubKey" type="string"/>
      </sequence>
    </complexType>
</element>
```

If the RI has not ever stored the public key of that OMA DRM v2.x Device, the RI MUST respond with *NoDevPubKey* error and SHOULD initiate a 4-pass registration protocol with that OMA DRM v2.x Device to acquire the public key of that OMA DRM v2.x Device. The RI could initiate the LRM to execute the LRM-RI DevPubKeyAcquisition protocol again by sending the LRM an LRM-RI DevPubKeyAcquisition trigger.

The other error codes of *Status* are specified in [SCE-GEN].

The DevPubKey carries the public key of OMA DRM v2.x Device. If *Status* contains any error, the DevPubKey field MUST NOT be present in the LRM-RIDevPubKeyAcquisitionResponse.

6.1.3 LRM-RI Create Device RO Protocol

The 2-pass LRM-RI Create Device RO protocol is the protocol by which an LRM enlists the services of an RI to Import Rights associated with some DRM Content Imported by the LRM from upstream DRM system to a designated OMA DRM v2.x Device, so that backward compatibility regarding Import function is achieved, i.e. an LRM can Import RO into an OMA DRM v2.x Device. This protocol assumes that the LRM and the OMA DRM v2.x Device each have a valid RI context for the associated RI.

This protocol includes secure transfer of Imported Rights and REK to the RI whereas ensures that the REK is not exposed to the RI. This protocol MAY involve OCSP protocol between the RI and an OCSP Responder for checking status of the RI's certificate chain. After successful 2-pass LRM-RI Create Device RO protocol execution, the RI MUST conduct RO Acquisition protocol including optional ROAP-ROAcquisition Trigger as per [DRM-DRM-v2.0], with the designated OMA DRM v2.x Device to issue the Imported-RO(s). But the RO Acquisition protocol itself is not part of this protocol.

Each LRM SHALL make sure that the number of recipient Devices is less than the threshold set by relevant upstream service providers. Such a threshold MAY vary over different upstream service providers, and MAY vary depending on the type of Import-Ready Data. Only in the case that the cumulative recipient Device quantity is less than the threshold does the LRM perform the LRM-RI Create Device RO protocol to issue Imported Rights to a recipient 2.x Device.

If the LRM has not acquired the public key of the OMA DRM v2.x Device before initiating the 2-pass LRM-RI Create Device RO protocol, LRM SHALL initiate the 2-pass LRM-RI DevPubKeyAcquisition protocol first.

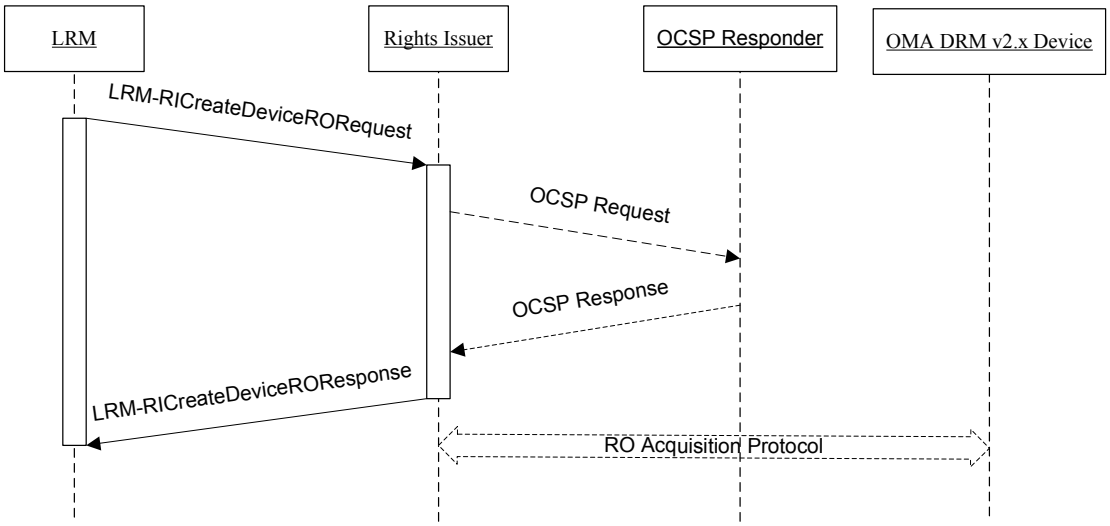

Figure 3 – The 2-pass LRM-RI Create Device RO Protocol

6.1.3.1 LRM-RI Create Device RO Request

An LRM sends the LRM-RI Create Device RO Request message to an RI to request the creation of one or more ROs for a designated OMA DRM v2.x Device. The root element of the message MUST be an <LRMRICreateDeviceRORequest> element of type gen:Request, in which the following elements are present:

Table 5: LRM-RICreateDeviceRORequest Message Parameters

element / attribute	usage	value
reqID	M	LRM's ID
resID	M	RI's ID
nonce	M	Default, as specified in [SCE-GEN]
time	M	Default, as specified in [SCE-GEN]
certificateChain	O	Default, as specified in [SCE-GEN]
reqInfo	M	Specified below
signature	M	Specified below

The <gen:reqInfo> element under the <LRMRICreateDeviceRORequest> element MUST contain an <lrm:LRMRICreateDeviceRORequestInformation> child element as defined by the following XML schema fragment:

```
<element name="LRMRICreateDeviceRORequest" type="gen:Request"/>
 <element name="LRMRICreateDeviceRORequestInformation">
  <complexType>
<sequence>
        <element name="sourceLRMID" type="gen:Identifier"/>
        <element name="recipientDeviceID" type="gen:Identifier"/>
        <!--multiple rights are used when Importing multiple RO to a single Device-->
<element name="rights" type="o-ex:rightsType" maxOccurs=" unbounded"/>
        <element name="encKeyInfo" type="xenc:EncryptedKeyType"/>
    <element name="mac" type="base64Binary"/>
  </sequence>
  </complexType>
 </element>
```

sourceLRMID: This element identifies the LRM which originated the request. It MUST contain the same value as the <reqID> element.

recipientDeviceID: This element identifies the OMA DRM v2.x Device to which the Rights will be Imported.

rights: This element conveys information about the Rights that the LRM is attempting to Import to the designated OMA DRM v2.x Device, including content ID, DCF Hash value, encrypted CEK, permissions and constraints. The corresponding elements within the <rights> element (specified in [DRM-REL-v2.1]) MUST be provided by the LRM. The element that holds ROID in the <rights> element MUST be present but with an arbitrary value. The RI SHALL replace the arbitrary value with a concrete ROID. For each RO being Imported, the LRM SHALL generate a random RO Encryption Key, K_{REK}, and use it to encrypt the CEK.

encKeyInfo: This element contains a wrapped concatenation of a MAC key, K_{MAC}, and one or more *encKey* values (see section 7.1.2 for details). For each RO being Imported, the LRM wraps the concatenation of K_{MAC} and K_{REK} with the recipient Device's public key. The wrapping result *encKey* is provided to the RI for forming the <encKey> element in the corresponding RO. The <ds:KeyInfo> element of <encKeyInfo> SHALL be the <gen:X509SPKIHash> element, identifying the RI public key through the (SHA-1) hash of the DER-encoded subjectPublicKeyInfo value in the RI certificate.

mac: This element provides integrity protection through a MAC on the canonical version ([SCE-GEN]) of the <reqInfo> element (excluding the <mac> element), using the K_{MAC} that is wrapped in the <encKeyInfo> element. The MAC algorithm SHALL be the same algorithm that was negotiated during the LRM-RI Registration, i.e. the MAC algorithm stored in the ResContext for the RI.

signature: This element contains a digital signature over the message besides the <signature> element itself. It is made using the negotiated signature algorithm and using the private key of the LRM.

Upon receiving the LRM-RICreateDeviceRORequest message, the RI MUST verify the signature of the LRM and check the freshness of the request by comparing the <nonce> value in the current request against retained previously received nonce values. It also checks the value of the <time> element in the request according to [DRM-DRM-v2.1]. If the LRM has an invalid DRM time, the RI must respond with *RequesterTimeError*. The RI then verifies that the MAC value contained in the <mac> element matches the content of <reqInfo>. It MUST also check whether the <sourceLRMID> element matches the signer of the request. Furthermore, the RI determines whether the cumulative number of recipient Devices serviced by the RI on behalf of the particular LRM would result in exceeding the upper bound set for the LRM. This upper bound is typically set by the RI directly or by a Trust Authority. If the upper bound would be exceeded, the RI SHALL reject the particular LRM-RICreateDeviceRORequest. In that event, later requests can still be accepted if they identify recipient Devices that are already on the list of recipient Devices for which the RI has provided ROs on behalf of the LRM. If verification of a request is successful and the RO(s) can be provided to the identified recipient Device without exceeding the upper bound, then the RI MUST issue RO(s) to the designated OMA DRM v2.x Device by a subsequent RO Acquisition protocol, based on the information about the Rights given by the <reqInfo> element.

The RI SHALL NOT include the <signature> element over the <rights> element in the RO Payload for the recipient Device.

6.1.3.2 LRM-RI Create Device RO Response

An RI sends the LRM-RI Create Device RO Response message to an LRM to indicate the result of creating one or more ROs for a designated OMA DRM v2.x Device. The root element of the message MUST be an <LRMRICreateDeviceROResponse> element of type gen:Response, in which the following elements are present:

Table 6: LRM-RICreateDeviceROResponse Message Parameters

element / attribute	usage	value
status	M	“Success”
reqID	M	LRM’s ID
resID	M	RI’s ID
nonce	M	Default, as specified in [SCE-GEN]
certificateChain	O	Default, as specified in [SCE-GEN]
ocspResponse	O	Default, as specified in [SCE-GEN]
resInfo	M	Specified below
signature	M	Specified below

The <gen:resInfo> element under the <LRMRICreateDeviceROResponse> element MUST contain an <lrm:LRMRICreateROResponseInformation> child element as defined by the following XML schema fragment:

```
<element name="LRMRICreateDeviceROResponse" type="gen:Response"/>
  <element name="LRMRICreateDeviceROResponseInformation">
    <complexType>
      <sequence maxOccurs="unbounded">
        <choice>
            <element name="success"/>
            <element name= "failureReason" type="string"/>
</choice>
      </sequence>
    </complexType>
</element>
```

resInfo: This element includes a sequence of choice between <success> element and <failureReason> element, one per RO being Imported. The first <success> or <failureReason> element gives the result for the RO corresponding to the first

<rights> element in the request; the second <success> or <failureReason> gives the result for the RO corresponding to the second <rights> element in the request, etc.

success: This element indicates that the corresponding <rights> element in the request was successfully processed and the corresponding RO can be issued to the designated OMA DRM v2.x Device.

failureReason: This element indicates the reason why the corresponding <rights> element in the request was not successfully processed.

signature: This element contains a digital signature over the message besides the <signature> element itself. It is made using the negotiated signature algorithm and using the private key of the RI.

6.1.4 LRM-RI Create Domain RO Protocol

The 2-pass LRM-RI Create Domain RO protocol is the protocol by which an LRM enlists the services of an RI to Import Rights associated with some DRM Content Imported by the LRM from upstream DRM system to a designated OMA DRM v2.x Domain, so that backward compatibility regarding Import function is achieved, i.e. an LRM can Import RO into an OMA DRM v2.x Domain. This protocol assumes that the LRM has a valid RI context for the associated RI.

This protocol includes mutual authentication of LRM and RI, secure transfer of Imported Rights and REK to the RI, and integrity-protected request and delivery of created Domain ROs. This protocol MAY involve OCSP protocol between the RI and an OCSP Responder for checking status of the RI's certificate chain. After receiving a newly created Domain RO from the RI, the LRM distributes it together with the corresponding Imported-Content to DRM Devices. But this is outside the scope of this specification.

Each LRM SHALL make sure that the number of recipient Domains is less than the threshold set by relevant upstream service providers. Such a threshold MAY vary over different upstream service providers, and MAY vary depending on the type of Import-Ready Data. Only in the case that the cumulative recipient Domain quantity is less than the threshold does the LRM perform the LRM-RI Create Domain RO protocol to issue Imported-Rights-Objects to a 2.x Domain.

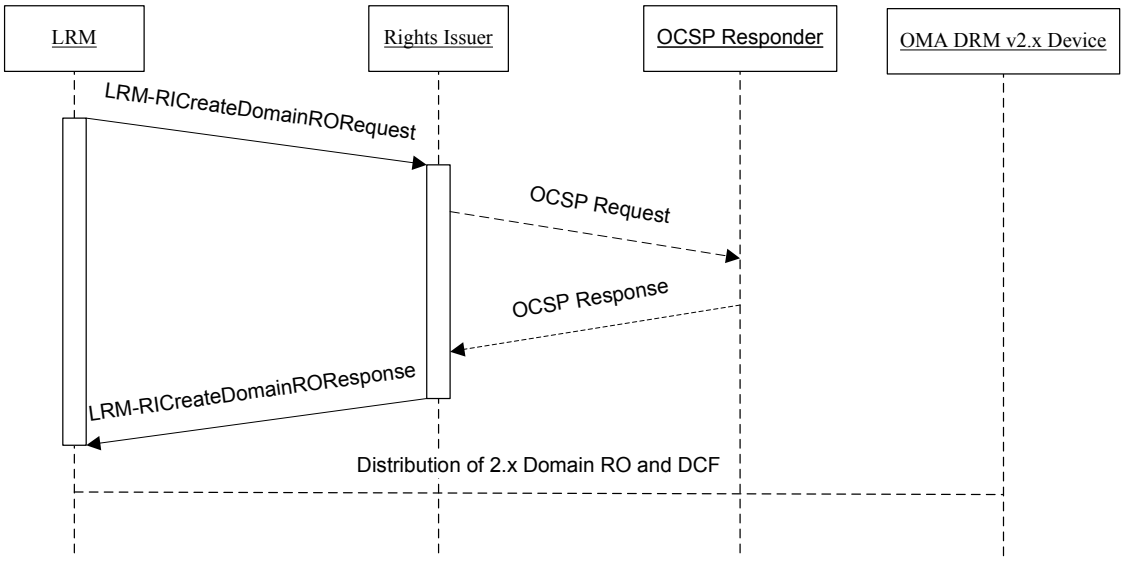

Figure 4 – The 2-pass LRM-RI Create Domain RO Protocol

6.1.4.1 LRM-RI Create Domain RO Request

An LRM sends the LRM-RI Create Domain RO Request message to an RI to request the creation of one or more domain ROs for a designated OMA DRM v2.x Domain. The root element of the message MUST be a <LRMRICreateDomainRORequest> element of type gen:Request, in which the following elements are present:

Table 7: LRM-RI Create Domain RO Request Message Parameters

element / attribute	Usage	value
reqID	M	LRM's ID
resID	M	RI's ID
nonce	M	Default, as specified in [SCE-GEN]
time	M	Default, as specified in [SCE-GEN]
certificateChain	O	Default, as specified in [SCE-GEN]
reqInfo	M	Specified below
signature	M	Specified below

The <gen:reqInfo> element under the <LRMRICreateDomainRORequest> element MUST contain an <lrm:LRMRICreateDomainRORequestInformation> child element as defined by the following XML schema fragment:

```
<element name="LRMRICreateDomainRORequest" type="gen:Request"/>
 <element name="LRMRICreateDomainRORequestInformation">
  <complexType>
<sequence>
        <element name="sourceLRMID" type="roap:Identifier"/>
        <element name="domainID" type="roap:DomainIdentifier"/>
<element name="rights" type="o-ex:rightsType" maxOccurs="unbounded"/>
        <element name="enc_REKs_Kmac" type="xenc:EncryptedKeyType"/>
    <element name="mac" type="base64Binary"/>
</sequence>
  </complexType>
 </element>
```

sourceLRMID: This element identifies the LRM which originated the request. It MUST contain the same value as the <reqID> element.

domainID: This element identifies the Domain to which the Rights will be Imported.

rights: This element conveys information about the Rights that the LRM is attempting to Import to the designated OMA DRM v2.x Domain, including content ID, DCF Hash value, encrypted CEK, permissions and constraints. The corresponding elements within the <rights> element (specified in [DRM-REL-v2.1]) MUST be provided by the LRM. The element that holds ROID in the <rights> element MUST be present but with an arbitrary value. The RI SHALL replace the arbitrary value with a concrete ROID. For each RO being Imported, the LRM SHALL generate a random RO Encryption Key, K_{REK}, and use it to encrypt the CEK.

encKeys: This element consists of a wrapped concatenation of a MAC key, K_{MAC}, and one or more K_{REK} (see section 7.1.3 for details). The child of the <ds:KeyInfo> element inside the <encKeys> element SHALL be the <roap:X509SPKIHash> element, identifying the RI's Public Key through the (SHA-1) hash of the DER-encoded subjectPublicKeyInfo value in the RI's Certificate.

mac: This element provides integrity protection through a MAC on the canonical version ([SCE-GEN]) of the <reqInfo> element (excluding the <mac> element) using the K_{MAC} which is wrapped in the <encKeys> element. The MAC algorithm SHALL be the same algorithm that was negotiated during LRM-RI Registration, i.e. the MAC algorithm stored in the ResContext for the RI.

signature: This element contains a digital signature over the message besides the <signature> element itself. It is made using the negotiated signature algorithm and using the private key of the LRM.

Upon receiving the LRM-RICreateDomainRORequest message, the RI MUST verify the signature of the LRM and check the freshness of the request by comparing the <nonce> value in the current request against retained previously received nonce values. It also checks the value of the <time> element in the request according to [DRM-DRM-v2.1]. If the LRM has an invalid DRM time, the RI must respond with *RequesterTimeError*. The RI then verifies that the MAC value contained in the <mac> element matches the content of <ReqInfo>. It MUST also check whether the <sourceLRMID> element matches the signer of the request. Furthermore, the RI determines whether the cumulative number of Domains serviced by the RI on behalf of the particular LRM would result in exceeding the upper bound set for the LRM. This upper bound is typically set by the RI directly or by a Trust Authority. If the upper bound would be exceeded, the RI SHALL reject the particular LRM-RICreateDomainRORequest. In that event, later requests can still be accepted if they identify Domains that are already on the list of Domains for which the RI has provided ROs on behalf of the LRM. If verification of a request is successful and the RO(s) can be provided to the identified Domain without exceeding the upper bound, then the RI MUST create Domain RO(s) based on the information about the Rights given by the <reqInfo> element.

6.1.4.2 LRM-RI Create Domain RO Response

An RI sends the LRM-RI Create Domain RO Response message to an LRM to indicate the result of creating one or more ROs for a designated OMA DRM v2.x Domain. The root element of the message MUST be an <LRMRICreateDomainROResponse> element of type gen:Response, in which the following elements are present:

Table 8: LRM-RI Create Domain RO Response Message Parameters

element / attribute	usage	Value
status	M	“Success”
reqID	M	LRM’s ID
resID	M	RI’s ID
nonce	M	Default, as specified in [SCE-GEN]
certificateChain	O	Default, as specified in [SCE-GEN]
ocspResponse	O	Default, as specified in [SCE-GEN]
resInfo	M	Specified below
signature	M	Specified below

The <gen:resInfo> element under the <LRMRICreateDomainROResponse> element MUST contain an <lrm:LRMRICreateDomainROResponseInformation> child element as defined by the following XML schema fragment:

```
<element name="LRMRICreateDomainROResponse" type="gen:Response"/>
 <element name="LRMRICreateDomainROResponseInformation">
  <complexType>
   <sequence maxOccurs="unbounded">
    <choice>
      <element name="domainRO" type="roap:ProtectedRO"/>
      <element name="failureReason" type="string"/>
</choice>
   </sequence>
  </complexType>
```

</element>

resInfo: This element includes a sequence of choice between <domainRO> element and <failureReason> element, one per RO being Imported. The first <domainRO> or <failureReason> element gives the result for the RO corresponding to the first <rights> element in the request; the second <domainRO> or <failureReason> gives the result for the RO corresponding to the second <rights> element in the request, etc.

domainRO: This element carries the newly created RO for the designated OMA DRM v2.x Domain. The LRM MUST verify that the <rights> element in the Domain RO matches the one it sent in the previous request message. If there is any inconsistency the LRM SHALL regard the Import operation as a failure and discard the Domain RO.

failureReason: This element indicates the reason why the corresponding <rights> element was not successfully processed.

signature: This element contains a digital signature over the message besides the <signature> element itself. It is made using the negotiated signature algorithm and using the private key of the RI.

6.1.5 Replay Cache Management for SCE-4-LRMP

This section describes how an LRM and an RI manages replay cache to prevent replay attack from malicious third party and to handle retry of request message from Source Device.

When an RI receives a request message for LRM-RI Create Device RO protocol or LRM-RI Create Domain RO protocol, it SHALL check the<reqID> and the <nonce> element in the request message as follows:

- If the <reqID> element and <nonce> element in the request message matches with one of replay cache entry, the RI SHALL check the <time> element in the request message as follows:
  - If the <time> element in the request message matches with a <time> in the replay cache entry, the RI SHALL ignore the request message.
  - Else, the RI SHALL generate a response message which contains parameters from response information in the matching replay cache entry, and send the generated response message, and then abort the process. The RI SHALL NOT continue to process the request message.
- Else, the RI SHALL create one replay cache entry which is at least composed of <reqID>, <nonce> and <time> which are copied from the request message, and continue to process the request message according to processing rules of each specific protocol (see section 6.1.3 and 6.1.4). During the processing, whenever the RI generates the response message, the RI SHALL store a response information additionally into the created replay cache entry, where the response information contains status, errorMessage, errorRedirectURL attributes and any relevant associated data in the response message.

It is strongly RECOMMENDED that the RI does not remove the replay cache entry until when the RI deems that the replay cache entry is old enough so that the LRM will no longer retry with the same nonce. In any case, the RI MUST keep the replay cache entry until the time when the RI would reject the request based on expiration, where the validity time window is RI implementation specific.

6.2 SCE-5-LRMP

The SCE-5-LRMP interface includes six protocols of which each is the same protocol that is used over the SCE-3-RDP interface as specified in [SCE-DOM] except that here the protocol is between the LRM and DEA rather than between the RI and DEA.

These six protocols are "RI-DEA Registration protocol", "Get User Domain Authorization protocol", "Drop User Domain Authorization protocol", "Proxy Join User Domain protocol", "Proxy Leave User Domain protocol" and "DEA Indirectly Triggers a v2.x DRM Agent to Leave a User Domain" (all as defined in [SCE-DOM]).

The LRM certificate MUST have the oma-kp-localRightsManagerDomain key purpose in order to perform any of the above six protocols. Additionally, for “Proxy Join User Domain protocol" or “ Proxy Leave User Domain protocol” or “DEA

Indirectly Triggers a v2.x DRM Agent to Leave a User Domain", the LRM certificate MUST have the oma-kp-rightsIssuer key purpose.

6.3 SCE-6-LRMP

6.3.1 Registration between a DRM Agent and an LRM

Before an LRM with the oma-kp-localRightsManagerDevice key purpose can generate a Device RO for a particular Device, the DRM Agent in that Device needs to register with the LRM.

Registration is done in the same way as the DRM Agent registers with the RI, i.e. over the 4-pass ROAP registration protocol (see [SCE-DRM]). The DRM Agent needs to perform the same procedures as when it registers with an RI, except that the DRM Agent MUST verify that the LRM has at least the oma-kp-localRightsManagerDevice key purpose.

In addition to checking for the presence of the oma-kp-drmAgent key purpose in the Device certificate, the LRM checks whether or not the oma-kp-sceDrmAgent key purpose is present. If an LRM does not have the oma-kp-rightsIssuer key purpose, it MUST reject registration if the oma-kp-sceDrmAgent key purpose is absent.

The DRM Agent distinguishes an LRM from an RI by its key purpose(s): if the entity has an oma-kp-localRightsManagerDevice key purpose, or an oma-kp-localRightsManagerDomain key purpose, the DRM agent knows that it is communicating with an LRM.

Registration results in a Device context on the LRM side, and an LRM Context on the DRM Agent side. The LRM Context is similar to the RI Context defined in [SCE-DRM]. From the LRM Context, the DRM Agent MUST be able to determine the LRM key purpose(s).

Since the DRM Agent distinguishes the LRM from an RI, if allowed by the trust model, the DRM Agent MAY determine revocation status of the LRM based on the use of CRLs rather than based on acquiring an OCSP response.

Since the DRM Agent is required to support DRM Time, the DRM Agent MAY acquire an OCSP response identifying the LRM from an entity other than that particular LRM. If allowed by the trust model, as long as such OCSP response has not expired, the DRM Agent MAY continue to successfully interact with the LRM.

If allowed by the trust model, an LRM MAY be exempt from supporting direct interaction with an OCSP Responder. If allowed by the trust model, an LRM MAY be exempt from providing OCSP Responses to DRM Agents.

6.3.2 Registration between an OMA DRM v2.x Agent and an LRM

Before an LRM with the oma-kp-rightsIssuer key purpose can generate a Device RO, a Domain RO or a User Domain RO upon request of a particular OMA DRM v2.x Device, the DRM Agent in that OMA DRM v2.x Device needs to register with the LRM. The registration of an OMA DRM v2.x Device (i.e. DRM Agent without additional oma-kp-sceDrmAgent key purpose) with an LRM with at least the oma-kp-rightsIssuer key purpose is identical to the registration of an OMA DRM v2.x Device with an RI from the Device's perspective.

In order to successfully interact with OMA DRM v2.x Devices, an LRM with the oma-kp-rightsIssuer key purpose MUST support timely access to an OCSP Responder. An LRM with the oma-kp-rightsIssuer key purpose MUST be trusted to reliably determine the need to acquire OCSP responses in order to correct DRM Time of OMA DRM v2.x Devices.

6.3.3 Generation and delivery of Imported Device ROs

After the DRM Agent has registered with the LRM, the LRM can generate and deliver Imported Device ROs to that DRM Agent. The delivery is performed via either the 2-pass ROAP RO Acquisition protocol or the 1-pass ROAP RO Delivery protocol (see [DRM-DRM-v2.1]). The 2-pass RO Acquisition protocol MAY be initiated by a ROAP RO Acquisition trigger.

To generate a Device RO for an SCE Device, the LRM MUST have at least an oma-kp-localRightsManagerDevice key purpose. To generate a Device RO for an OMA DRM v2.x Device the LRM MUST have at least the oma-kp-rightsIssuer key purpose. An LRM with no oma-kp-rightsIssuer key purpose MAY use the LRM-RI Create Device RO protocol.

The DRM Agent SHALL NOT consume an RO before it has verified that the RI/LRM that created the RO was allowed to do so. For this, the DRM Agent MUST check the key purpose of the RI/LRM in the associated RI/LRM context.

6.3.4 Import into OMA DRM v2.x Domains

Because OMA DRM v2.x Domains are normally managed by an RI, an LRM with the oma-kp-rightsIssuer key purpose MAY manage an OMA DRM v2.x Domain itself. For managing an OMA DRM v2.x Domain, the LRM MUST have at least the oma-kp-rightsIssuer key purpose. In this case, the LRM MUST have a Device context for each OMA DRM v2.x Device in the Domain. For this, it MUST have performed the Registration protocol as specified in [DRM-DRM-v2.1]. The OMA DRM v2.x Domain is managed using the regular ROAP JoinDomain and LeaveDomain protocols (see [DRM-DRM-v2.1]), where the LRM performs the tasks of the RI.

An LRM with no oma-kp-rightsIssuer key purpose (i.e. with only an oma-kp-localRightsManagerDomain and/or an oma-kp-localRightsManagerDevice key purpose) MUST NOT manage its own OMA DRM v2.x Domains. An LRM with no oma-kp-rightsIssuer key purpose MAY use the LRM-RI Create Domain RO protocol.

For an LRM without the oma-kp-localRightsManagerDevice key purpose to Import to (SCE conformant) Devices, these Devices should be included in a User Domain associated to this LRM (see [SCE-DOM]).

7. Key Management

7.1 Key Transport Mechanisms

7.1.1 Import Protocol

A DRM Agent can Import from an LRM under an Import protocol which is indicated as RO Acquisition protocol [SCE-DRM], or 2-pass Rights Object Acquisition Protocol [DRM-DRM-v2.1] or 1-pass Rights Object Acquisition Protocol [DRM-DRM-v2.1].

For the 1-pass or 2-pass Rights Object Acquisition Protocol in [DRM-DRM-v2.1], the LRM certificate SHALL have the oma-kp-rightsIssuer key purpose. For the RO Acquisition protocol in [SCE-DRM], the LRM certificate SHALL have the oma-kp-localRightsManagerDevice key purpose or the oma-kp-localRightsManagerDomain key purpose.

7.1.2 Transporting KMAC and one or more encKey under an RI Public Key

This section applies to the LRM-RI Create Device RO protocol.

encKey is the result of wrapping K_{MAC} and K_{REK} with the recipient Device's RSA public key, using RSAES-KEM-KWS and AES_WRAP [DRM-DRM-v2.1]. For the AES-WRAP scheme, K_{MAC} and K_{REK} are concatenated to form K, i.e.:

$KEK_i = KDF(I2OSP(Z_i, mLen_{Device}), \textbf{NULL}, kekLen)$

$C_{i2} = AES\text{-}WRAP(KEK_i, K_{MAC} \mid K_{REKi})$

$C_{i1} = I2OSP(RSA.ENCRYPT(PubKey_{Device}, Z_i), mLen_{Device})$

$encKey_i = C_{i1} \mid C_{i2}$

To securely transmit K_{MAC} and the calculated encKey values (one per RO being Imported) to the RI, RSAES-KEM-KWS SHALL be used with AES-WRAP. For the AES-WRAP scheme, K_{MAC} and the encKey values are concatenated to form K, i.e.:

$KEK = KDF(I2OSP(Z, mLen_{RI}), \textbf{NULL}, kekLen)$

$K = K_{MAC} \mid encKey_1 \mid \ldots \mid encKey_n$ (*n* is the number of ROs being Imported; the enckey values are concatenated in the same order as the corresponding <rights> elements appear in the request)

$C_2 = AES\text{-}WRAP(KEK, K)$

$C_1 = I2OSP(RSA.ENCRYPT(PubKey_{RI}, Z), mLen_{RI})$

$C = C_1 \mid C_2$

After receiving C, the RI splits it into C_1 and C_2 and decrypts C_1 using its private key (consisting of a private exponent d and the modulus m), yielding Z:

$C_1 \mid C_2 = C$

$c_1 = OS2IP(C_1, mLen)$

$Z = RSA.DECRYPT(PrivKey_{RI}, c_1)$

where the function OS2IP() is the same as specified in [DRM-DRM-v2.1].

Using Z, the RI can derive KEK, and from KEK unwrap C2 to yield K_{MAC} and $encKey_1, \ldots, encKey_n$:

$KEK = KDF(I2OSP(Z, mLen), NULL, kekLen)$

$K_{MAC} \mid encKey_1 \mid \ldots \mid encKey_n = AES\text{-}UNWRAP(KEK, C2)$

The $encKey_i$ is used by the RI to form the <encKey> element in <roPayload> in subsequent RO Acquisition protocol.

7.1.3 Transporting KMAC and one or more KREK under an RI Public Key

This section applies to the LRM-RI Create Domain RO protocol.

K_{MAC} and K_{REK} are each 128-bit long keys generated randomly by the LRM. K_{REK} ("Rights Object Encryption Key") is the wrapping key for the content-encryption key K_{CEK} in Rights Objects. K_{MAC} is used for key confirmation of the message carrying one or more K_{REK}.

The asymmetric encryption scheme RSAES-KEM-KWS SHALL be used with the AES-WRAP symmetric-key wrapping scheme to securely transmit K_{MAC} and one or more K_{REK} to a recipient RI using the RI's RSA public key. An independent random value Z SHALL be chosen for each encryption operation. For the AES-WRAP scheme, K_{MAC} and one or more K_{REK} are concatenated to form K, i.e.:

KEK = KDF(I2OSP(Z, mLen), **NULL**, kekLen)

C_2 = AES-WRAP(KEK, K_{MAC} | K_{REK1} | … | K_{REKn}) (*n* is the number of ROs being Imported; the K_{REK}'s are concatenated in the same order as the corresponding <rights> elements appear in the request)

C_1 = I2OSP(RSA.ENCRYPT($PubKey_{RI}$, Z), mLen)

C = C_1 | C_2

where kekLen SHALL be set to 16 (128 bits) and mLen is the length of the modulus of the RI's RSA public key in octets. In this way, AES-WRAP is used to wrap 128 * (n + 1) bits of key data (K_{MAC} | K_{REK1} | … | K_{REKn}) with a 128-bit key-encryption key (KEK).

After receiving C, the RI splits it into C_1 and C_2 and decrypts C_1 using its private key (consisting of a private exponent d and the modulus m), yielding Z:

C_1 | C_2 = C

c_1 = OS2IP(C_1, mLen)

Z = RSA.DECRYPT($PrivKey_{RI}$, c_1) = c_1^d mod m

where OS2IP converts an octet string to a nonnegative integer and is defined in PKCS #1 (see [RFC3447]).

Using Z, the RI can derive KEK, and from KEK unwrap C_2 to yield K_{MAC} and K_{REK1}, …, K_{REKn}.:

KEK = KDF(I2OSP(Z, mLen), **NULL**, kekLen)

K_{MAC} | K_{REK1} | … | K_{REKn} = AES-UNWRAP(KEK, C_2)

The following URI SHALL be used to identify this key transport scheme in **<xenc:EncryptionMethod>** elements:

http://www.rsasecurity.com/rsalabs/pkcs/schemas/pkcs-1#rsaes-kem-kdf2-kw-aes128

7.2 Certificate Handling

The certificate profiles are specified in Appendix C. The LRM certificate profile ensures that LRM certificates are distinguishable from DRM Agent/Device certificates and from RI certificates by SCE-conformant Devices. This is achieved by mandating inclusion of the **`oma-kp-localRightsManagerDevice`** or **`oma-kp-localRightsManagerDomain`** key purpose in the certificate for the LRM.

The LRM certificate profile allows to ensure that LRM certificates are distinguishable from RI certificates by OMA DRM Devices that are not conformant to SCE in that such LRM certificates will be rejected for effective use by DRM Agents in such SCE- non-conformant Devices. This is achieved through exclusion of the **`oma-kp-rightsIssuer`** key purpose from the certificate.

The LRM certificate profile also allows LRM certificates to be indistinguishable from RI certificates by OMA DRM Devices that are not conformant to SCE, such that LRM certificates can be used as RI certificates by DRM Agents in such SCE- non-conformant Devices. This is achieved through inclusion of the **`oma-kp-rightsIssuer`** key purpose in the certificate.

8. Security Considerations (Informative)

In order to cryptographically enforce the intended restrictions on information flow of the LRM-RI Create Device RO protocol, it is necessary that REK values are not exposed to the RI during or subsequent to execution of this protocol. If all of the following measures are taken, then this is achieved:

- The supporting LRM-RI DevPubKeyAcquisition protocol is modified to return the 2.x DRM Agent certificate chain (rather than just a public key that is purported to correspond to a 2.x DRM Agent) and/or the LRM restricts its use of the LRM-RI Create Device RO protocol to those intended recipient 2.x DRM Agents for which it has access to the entity certificate;
- The LRM verifies that the public key that it uses to encrypt the REK for use in the LRM-RI Create Device RO protocol corresponds to a certificate that includes an oma-kp-drmAgent key purpose and does not include an oma-kp-sceDrmAgent key purpose;
- The RI does not use the LRM-RI Create Device RO protocol to generate ROs for entities other than 2.x DRM Agents (as determined by the entity certificate or existing context for the entity)
    - In particular, the RI does not use this protocol to generate ROs for SCE DRM Agents or other RIs;
- The RI does not accept requests to use the Move Device RO via RI protocol [SCE-DRM] from entities other than SCE DRM Agents (as determined by the entity certificate or existing context for the entity), i.e., the RI does not use its private key to recover REKs corresponding to MoveDeviceRORequest messages from entities other than SCE DRM Agents.
    - This precludes, in particular, a 2.x DRM Agent from successfully submitting a MoveDeviceRORequest message that exposes REKs to the RI from ROs received as a consequence of an LRM performing the LRM-RI Create Device RO protocol.

Appendix A. Change History (Informative)

A.1 Approved Version History

Reference	Date	Description
OMA-TS-SCE_LRM-V1_0-20110705-A	05 Jul 2011	Status changed to Approved by TP: OMA-TP-2011-0233-INP_SCE_V1_0_ERP_for_Final_Approval

Appendix B. Static Conformance Requirements (Normative)

The notation used in this appendix is specified in [SCR-RULES].

B.1 SCR for DRM Agent

Item	Function	Reference	Requirement
SCE-LRM-DRMAGENT-C-001-O	Registration between DRM Agent and LRM	6.3.1, [SCE-DRM]	
SCE-LRM-DRMAGENT-C-002-O	Import protocol (Device RO)	6.3.3, 7.1.1, [SCE-DRM]	
SCE-LRM-DRMAGENT-C-003-O	Import protocol (User Domain RO)	7.1.1, [SCE-DRM]	

B.2 SCR for RI

Item	Function	Reference	Requirement
SCE-LRM-RI-S-001-O	LRM-RI Registration protocol	6.1.1	
SCE-LRM-RI-S-002-O	LRM-RI DevPubKeyAcquisition protocol	6.1.2	SCE-LRM-RI-S-001-O
SCE-LRM-RI-S-003-O	LRM-RI Create Device RO protocol	6.1.3	SCE-LRM-RI-S-001-O AND SCE-LRM-RI-S-002-O AND SCE-LRM-RI-S-005-O
SCE-LRM-RI-S-004-O	LRM-RI Create Domain RO protocol	6.1.4	SCE-LRM-RI-S-001-O AND SCE-LRM-RI-S-006-O
SCE-LRM-RI-S-005-O	Transporting KMAC and one or more encKey under an RI Public Key	7.1.2	
SCE-LRM-RI-S-006-O	Transporting KMAC and one or more KREK under an RI Public Key	7.1.3	

B.3 SCR for DEA

Item	Function	Reference	Requirement
SCE-LRM-DEA-S-001-M	Join User Domain protocol	6.2, [SCE-DOM]	
SCE-LRM-DEA-S-002-M	Leave User Domain protocol	6.2, [SCE-DOM]	

Item	Function	Reference	Requirement
SCE-LRM-DEA-S-003-M	Registration between RI and DEA	6.2, [SCE-DOM]	
SCE-LRM-DEA-S-004-M	Get User Domain Authorization protocol	6.2, [SCE-DOM]	
SCE-LRM-DEA-S-005-M	Drop User Domain Authorization protocol	6.2, [SCE-DOM]	
SCE-LRM-DEA-S-006-O	Proxy Join User Domain protocol	6.2, [SCE-DOM]	SCE-LRM-DEA-S-007-O
SCE-LRM-DEA-S-007-O	Proxy Leave User Domain protocol	6.2, [SCE-DOM]	SCE-LRM-DEA-S-008-O
SCE-LRM-DEA-S-008-O	Proxy Leave User Domain Trigger	6.2, [SCE-DOM]	

B.4 SCR for LRM with at least LRM-Device key purpose but without RI key purpose (LRMDEV or LRMDEV/LRMDOM)

Item	Function	Reference	Requirement
SCE-LRM-LRMDEV-S-001-O	Registration between DRM Agent and LRM	6.3.1, [SCE-DRM]	
SCE-LRM-LRMDEV-S-002-O	Import protocol (Device RO)	6.3.3, 7.1.1, [SCE-DRM]	SCE-LRM-LRMDEV-S-001-O
SCE-LRM-LRMDEV-S-003-O	LRM-RI Registration protocol	6.1.1	
SCE-LRM-LRMDEV-S-004-O	LRM-RI DevPubKeyAcquisition protocol	6.1.2	SCE-LRM-LRMDEV-S-003-O
SCE-LRM-LRMDEV-S-005-O	LRM-RI Create Device RO protocol	6.1.3	SCE-LRM-LRMDEV-S-003-O AND SCE-LRM-LRMDEV-S-004-O AND SCE-LRM-LRMDEV-S-007-O
SCE-LRM-LRMDEV-S-006-O	LRM-RI Create Domain RO protocol	6.1.4	SCE-LRM-LRMDEV-S-003-O AND SCE-LRM-LRMDEV-S-008-O
SCE-LRM-LRMDEV-S-007-O	Transporting KMAC and one or more encKey under an RI Public Key	7.1.2	
SCE-LRM-LRMDEV-S-008-O	Transporting KMAC and one or more KREK under an RI Public Key	7.1.3	

B.5 SCR for LRM with at least LRM-Domain key purpose but without RI key purpose (LRMDOM or LRMDEV/LRMDOM)

Item	Function	Reference	Requirement
SCE-LRM-LRMDOM-S-001-O	Registration between DRM Agent and LRM	6.3.1, [SCE-DRM]	
SCE-LRM-LRMDOM-S-002-O	Import protocol (User Domain RO)	7.1.1, [SCE-DRM]	SCE-LRM-LRMDOM-S-001-O *(for <userDomain>-constrained ROs)* AND SCE-LRM-LRMDOM-S-007-O AND SCE-LRM-LRMDOM-S-008-O AND SCE-LRM-LRMDOM-S-009-O
SCE-LRM-LRMDOM-S-003-O	LRM-RI Registration protocol	6.1.1	
SCE-LRM-LRMDOM-S-004-O	LRM-RI DevPubKeyAcquisition protocol	6.1.2	SCE-LRM-LRMDOM-S-003-O
SCE-LRM-LRMDOM-S-005-O	LRM-RI Create Device RO protocol	6.1.3	SCE-LRM-LRMDOM-S-003-O AND SCE-LRM-LRMDOM-S-004-O AND SCE-LRM-LRMDOM-S-0010-O
SCE-LRM-LRMDOM-S-006-O	LRM-RI Create Domain RO protocol	6.1.4	SCE-LRM-LRMDOM-S-003-O AND SCE-LRM-LRMDOM-S-011-O
SCE-LRM-LRMDOM-S-007-O	RI-DEA Registration protocol	6.2, [SCE-DOM]	
SCE-LRM-LRMDOM-S-008-O	Get User Domain Authorization protocol	6.2, [SCE-DOM]	
SCE-LRM-LRMDOM-S-009-O	Drop User Domain Authorization protocol	6.2, [SCE-DOM]	
SCE-LRM-LRMDOM-S-0010-O	Transporting KMAC and one or more encKey under an RI Public Key	7.1.2	
SCE-LRM-LRMDOM-S-0011-O	Transporting KMAC and one or more KREK under an RI Public Key	7.1.3	

B.6 SCR for LRM with RI key purpose and at least LRM-Domain key purpose (LRMDOM/RI or LRMDEV/DOM/RI)

Item	Function	Reference	Requirement
SCE-LRM-LRMDOM/RI-S-001-O	Registration between DRM Agent and LRM	6.3.1, [SCE-DRM]	
SCE-LRM-LRMDOM/RI-S-002-O	Registration between v2.x DRM Agent and LRM	6.3.2, [DRM-DRM-v2.1]	
SCE-LRM-LRMDOM/RI-S-003-O	Import protocol (User Domain RO)	7.1.1, [SCE-DRM]	SCE-LRM-LRMDOM/RI-S-001-O *(for <userDomain>-constrained*

Item	Function	Reference	Requirement
			ROs) AND SCE-LRM-LRMDOM/RI-S-007-O AND SCE-LRM-LRMDOM/RI-S-008-O AND SCE-LRM-LRMDOM/RI-S-009-O AND SCE-LRM-LRMDOM/RI-S-010-O *(for backwards-compatible RO)* AND SCE-LRM-LRMDOM/RI-S-011-O *(for backwards-compatible RO)* AND SCE-LRM-LRMDOM/RI-S-012-O *(for backwards-compatible RO)*
SCE-LRM-LRMDOM/RI-S-004-O	Import protocol (into v2.x Domain)	6.3.4, 7.1.1, [DRM-DRM-v2.1]	
SCE-LRM-LRMDOM/RI-S-005-O	Import protocol (Device RO for v2.x DRM Agent)	6.3.3, 7.1.1, [DRM-DRM-v2.1]	SCE-LRM-LRMDOM/RI-S-002-O
SCE-LRM-LRMDOM/RI-S-006-O	LRM-RI Registration protocol	6.1.1; *Enables use of Move Domain RO protocol, with LRM as original Issuer and RI designated within <moveIndication> element* [SCE-DRM], [SCE-REL]	
SCE-LRM-LRMDOM/RI-S-007-O	RI-DEA Registration protocol	6.2, [SCE-DOM]	
SCE-LRM-LRMDOM/RI-S-008-O	Get User Domain Authorization protocol	6.2, [SCE-DOM]	
SCE-LRM-LRMDOM/RI-S-009-O	Drop User Domain Authorization protocol	6.2, [SCE-DOM]	
SCE-LRM-LRMDOM/RI-S-010-O	Proxy Join User Domain protocol	6.2, [SCE-DOM]	
SCE-LRM-LRMDOM/RI-S-011-O	Proxy Leave User Domain protocol	6.2, [SCE-DOM]	
SCE-LRM-LRMDOM/RI-S-012-O	DEA Indirectly Triggers a v2.x DRM Agent to Leave a User Domain protocol	6.2, [SCE-DOM]	

B.7 SCR for LRM with RI key purpose and at least LRM-Device key purpose (LRMDEV/RI or LRMDEV/LRMDOM/RI)

Item	Function	Reference	Requirement
SCE-LRM-LRMDEV/RI-S-001-O	Registration between DRM Agent and LRM	6.3.1, [SCE-DRM]	
SCE-LRM-LRMDEV/RI-S-002-O	Registration between v2.x DRM Agent and LRM	6.3.2, [DRM-DRM-v2.1]	
SCE-LRM-LRMDEV/RI-S-003-O	Import protocol (into v2.x Domain)	6.3.4, 7.1.1, [DRM-DRM-v2.1]	
SCE-LRM-LRMDEV/RI-S-004-O	Import protocol (Device RO for v2.x DRM Agent)	6.3.3, 7.1.1, [DRM-DRM-v2.1]	SCE-LRM-LRMDEV/RI-S-002-O
SCE-LRM-LRMDEV/RI-S-005-O	LRM-RI Registration protocol	6.1.1; *Enables use of Move Device RO protocol, with LRM as original Issuer and RI designated within <moveIndication> element* [SCE-DRM], [SCE-REL]	

Appendix C. Certificate Profiles (Normative)

C.1 LRM Certificates

The profile for LRM certificates follows the profile for "X.509-compliant server certificate" in [CERT-PROF] with the following modifications:

Signature	MUST be RSA with SHA-1
Serial Number	MUST be less than, or equal to, 20 bytes in length
Issuer Name	MUST be present and MUST use a subset of the following naming attributes from [CERT-PROF] – countryName, organizationName, organizationalUnitName, commonName, and stateOrProvinceName.
Subject Name	MUST be present and MUST use a subset of the following naming attributes from [CERT-PROF] – countryName, organizationName, organizationalUnitName, commonName, and serialNumber. The structure and contents of a Device subject name shall be as follows: [countryName=<Country of manufacturer>] [organizationName=<Manufacturer company name>] [organizationalUnitName=<Manufacturing location>] [commonName=<Model name>] serialNumber=<Unique identifier for Device, as assigned by the Certificate Issuer. Does not have to be the same as the IMEI> The serialNumber attribute MUST be present. The countryName, organizationName, organizationalUnitName, and commonName may be present. Other attributes are not allowed and must not be included. For all naming attributes of type DirectoryString, the PrintableString or the UTF8String choice must be used. Note that the maximum length (in octets) for values of these attributes is as follows: countryName – 2 (country code in accordance with ISO/IEC 3166), organizationName, organizationalUnitName, commonName, and serialNumber – 64. Example: C="US";O="DRM Devices 'R Us"; CN="DRM Device Mark VI"; SN="1234567890"
Extensions	The extKeyUsage extension SHALL be present, and contain (at least) the **oma-kp-localRightsManagerDevice** or the **oma-kp-localRightsManagerDomain** key purpose object identifier: oma-kp-localRightsManagerDevice OBJECT IDENTIFIER ::= {oma-kp 7} oma-kp-localRightsManagerDomain OBJECT IDENTIFIER ::= {oma-kp 8} CAs MUST set this extension to critical. If the keyUsage extension is present (recommended), then the digitalSignature bit shall be set. When present, this extension shall be set to critical. CAs MAY include the certificatePolicy extension, indicating the policy the certificate has been issued under, and possibly containing a URI identifying a source of more information about the policy.

	CAs are recommended to not include any other extensions, but may, for compliance with [RFC3280], include the authorityKeyIdentifier extension. CAs MUST NOT include any other critical extensions.

SCE DRM Agents processing LRM certificates MUST meet the requirements on clients processing "X.509-compliant server certificates" defined in [CERT-PROF]. In addition, SCE DRM Agents:

- MUST be able to process LRM certificates up to 1500 bytes long;
- MUST be able to process LRM certificates with serial numbers 20 bytes long; and
- MUST recognize the presence of the **oma-kp-localRightsManagerDevice** and **oma-kp-localRightsManagerDomain** object identifiers defined above in the extKeyUsage extension in LRM certificates. If one (or both) of these is present, then the SCE DRM Agent MUST consider the subject certified by the certificate to be a LRM while processing information received from it.

Note: If the **oma-kp-rightsIssuer** object identifier defined in [DRM-DRM-v2.1] for the extKeyUsage extension is present *in addition* to the **oma-kp-localRightsManagerDevice** or **oma-kp-localRightsManagerDomain** defined above, then the SCE DRM Agent MUST still consider the subject certified by the certificate to be an LRM while processing information received from it.

C.2 CA Certificates

The CA certificates for use by SCE follow the OMA DRM CA certificate profile specified in [DRM-v2.1]. In addition, the following requirements apply:

LRMs and SCE DRM Agents MUST meet the requirements on relying parties defined in [CERT-PROF]. Note that this implies, among other things, a requirement on LRMs and SCE DRM Agents to also recognize the basicConstraints and the subjectKeyIdentifier extensions.

Appendix D. Message Examples (Informative)

D.1 LRMRIRegistrationTrigger

```
<?xml version="1.0" encoding="UTF-8"?>
<gen:drmTrigger
 type="LRMRIRegistrationTrigger"
 version="1.0"
 xmlns:ds="http://www.w3.org/2000/09/xmldsig#"
 xmlns:gen="urn:oma:xml:sce:gen"
 xmlns:xenc="http://www.w3.org/2001/04/xmlenc#"
 xmlns:lrm="urn:oma:drm:sce:lrm"
 xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance">
<body id="idvalue0">
  <resID>
    <keyIdentifier xsi:type=“gen:X509SPKIHash”>
      <hash>aXENc+Um/9/NvmYKiHDLaErK0fk=</hash>
    </keyIdentifier>
  </resID>
  <nonce>c2FtcGxlVHJpZ2dlck5vbmNl</nonce>
  <reqURL>http://ri.example.com/ro.cgi?tid=qw683hgew7d</reqURL>
  <trgInfo>
      <lrm:LRMRIRegistrationTriggerInformation>
          <lrm:LRMID>n8yu98hy0e2109eu09ewf09u</lrm:LRMID>
      </lrm:LRMRIRegistrationTriggerInformation>
  </trgInfo>
 </body>
</gen:drmTrigger>
```

D.2 LRM-RIhelloRequest

```
<?xml version="1.0" encoding="UTF-8"?>
<gen:helloRequest>
 type="LRMRIRegistrationRequest"
 version="1.0"
 xmlns:ds="http://www.w3.org/2000/09/xmldsig#"
 xmlns:gen="urn:oma:xml:sce:gen"
 xmlns:xenc="http://www.w3.org/2001/04/xmlenc#"
 xmlns:lrm="urn:oma:drm:sce:lrm"
 xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance">
<gen:RegReqInfo">
   <trustedAuthorities>
      <keyIdentifier xsi:type="roap:X509SPKIHash">
        <hash>bew3e332oihde9dwiHDLaErK0fk=</hash>
      </keyIdentifier>
      <keyIdentifier xsi:type="roap:X509SPKIHash">
        <hash>3lkpoi9fceoiuoift45epokifc0poiss</hash>
      </keyIdentifier>
   </trustedAuthorities>
      <gen:serverInfo>bew3e332oihde9dwiHDLaErK0fk=</gen:serverInfo>
      <gen:deviceDetails>
        <gen:manufacturer> ABC</gen:manufacturer>
        <gen:model>abc</gen:model>
        <gen:version>1.0</gen:version>
     </gen:deviceDetails>
</gen:RegReqInfo">
</gen:helloRequest>
```

D.3 LRM-RIhelloResponse

```
<?xml version="1.0" encoding="UTF-8"?>
<gen:helloResponse>
 type="LRMRIRegistrationResponse"
 version="1.0"
 xmlns:ds="http://www.w3.org/2000/09/xmldsig#"
 xmlns:gen="urn:oma:xml:sce:gen"
 xmlns:xenc="http://www.w3.org/2001/04/xmlenc#"
 xmlns:lrm="urn:oma:drm:sce:lrm"
 xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance">
 <gen:RegResInfo>
  <resURL>http://ri.example.com/roap.cgi</resURL>
   <domainNameWhitelist>
    <dn>Home</dn>
    <dn>Office</dn>
  </DomainNameWhiteList>
 </gen:RegResInfo>
</gen:helloResponse>
```

D.4 LRM-RIRegistrationRequest

```
<?xml version="1.0" encoding="UTF-8"?>
<lrm:LRMRIRegistrationRequest
sessionId="433213"
 triggerNonce="c2FtcGxlVHJpZ2dlck5vbmNl"
 xmlns:ds="http://www.w3.org/2000/09/xmldsig#"
 xmlns:gen="urn:oma:xml:sce:gen"
 xmlns:xenc="http://www.w3.org/2001/04/xmlenc#"
 xmlns:lrm="urn:oma:drm:sce:lrm"
 xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance">
 <reqID>
  <keyIdentifier xsi:type="gen:X509SPKIHash">
   <hash>c2FtcGxlRFJNQWdlbnQ=</hash>
  </keyIdentifier>
 </reqID>
 <resID>
  <keyIdentifier xsi:type="gen:X509SPKIHash">
   <hash>c2FtcGxlUmlnaHRzSXNzdWVy</hash>
  </keyIdentifier>
 </resID>
 <nonce>RFJNQWdlbnROb25jZTA=</nonce>
 <time>2010-12-31T12:00:00</time>
<reqInfo>
   <lrm:LRMRIRegistrationRequestInformation>
    <supportedUpstreamDRMs>XXX</supportedUpstreamDRMs>
  <needMoveService/>
 </lrm:LRMRIRegistrationRequestInformation>
 </reqInfo>
 <signature>SignatureValue</signature>
</lrm:LRMRIRegistrationRequest>
```

D.5 LRM-RIRegistrationResponse

```
<?xml version="1.0" encoding="UTF-8"?>
<lrm:LRMRIRegistrationResponse
 sessionId="433213"
 status="Success"
```

```
  xmlns:ds="http://www.w3.org/2000/09/xmldsig#"
  xmlns:gen="urn:oma:xml:sce:gen"
  xmlns:xenc="http://www.w3.org/2001/04/xmlenc#"
  xmlns:lrm="urn:oma:drm:sce:lrm"
  xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance">
    <reqID>
    <keyIdentifier xsi:type="gen:X509SPKIHash">
      <hash>c2FtcGxlRFJNQWdlbnQ=</hash>
    </keyIdentifier>
  </reqID>
  <resID>
    <keyIdentifier xsi:type="gen:X509SPKIHash">
      <hash>c2FtcGxlUmlnaHRzSXNzdWVy</hash>
    </keyIdentifier>
  </resID>
  <nonce>RFJNQWdlbnROb25jZTA=</nonce>
<rspInfo>
   <lrm:LRMRIRegistrationResponseInformation>
       <selectedUpstreamDRMs>XXX </selectedUpstreamDRMs>
       <provideMoveService/>
   </lrm:LRMRIRegistrationResponseInformation>
  </rspInfo>
  <signature>SignatureValue</signature>
</lrm:LRMRIRegistrationResponse>
```

D.6 LRM-RIDevPubKeyAcquisitionTrigger

```
<?xml version="1.0" encoding="UTF-8"?>
<gen:drmTrigger
  type="LRMRIDevPubKeyAcquisitionTrigger"
  version="1.0"
  xmlns:ds="http://www.w3.org/2000/09/xmldsig#"
  xmlns:gen="urn:oma:xml:sce:gen"
  xmlns:xenc="http://www.w3.org/2001/04/xmlenc#"
  xmlns:lrm="urn:oma:drm:sce:lrm"
xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance">
<body id="idvalue1">
    <nonce>c2FtcGxlVHJpZ2dlck5vbmNl</nonce>
    <trgInfo>
       <lrm:LRMRIDevPubKeyAcquisitionTriggerInformation>
          <DevID> n8yu13540e2109eu09ewf09u=</DevID>
       </lrm:LRMRIDevPubKeyAcquisitionTriggerInformation>
    </trgInfo>
  </body>
</gen:drmTrigger>
```

D.7 LRM-RIDevPubKeyAcquisitionRequest

```
<?xml version="1.0" encoding="UTF-8"?>
<lrm:DevPubKeyAcquisitionRequest
  triggerNonce="c2FtcGxlVHJpZ2dlck5vbmNl"
 xmlns:ds="http://www.w3.org/2000/09/xmldsig#"
  xmlns:gen="urn:oma:xml:sce:gen"
  xmlns:xenc="http://www.w3.org/2001/04/xmlenc#"
  xmlns:lrm="urn:oma:drm:sce:lrm"
```

```
  xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance">
  <reqID>
    <keyIdentifier xsi:type="gen:X509SPKIHash">
      <hash>c2FtcGxlRFJNQWdlbnQ=</hash>
    </keyIdentifier>
  </reqID>
  <resID>
    <keyIdentifier xsi:type="gen:X509SPKIHash">
      <hash>c2FtcGxlUmlnaHRzSXNzdWVy</hash>
    </keyIdentifier>
  </resID>
  <nonce>RFJNQWdlbnROb25jZTA=</nonce>
  <time>2010-12-31T12:00:00</time>
<reqInfo>
     <lrm:LRMRIDevPubKeyAcquisitionRequestInformation>
        <DevID> n8yu13540e2109eu09ewf09u=</DevID>
     </lrm:LRMRIDevPubKeyAcquisitionRequestInformation>
  </reqInfo>
  <signature>SignatureValue</signature>
</lrm:DevPubKeyAcquisitionRequest>
```

D.8 LRM-RIDevPubKeyAcquisitionResponse

```
<?xml version="1.0" encoding="UTF-8"?>
<lrm:DevPubKeyAcquisitionResponse
  status="Success"
  xmlns:ds="http://www.w3.org/2000/09/xmldsig#"
  xmlns:gen="urn:oma:xml:sce:gen"
  xmlns:xenc="http://www.w3.org/2001/04/xmlenc#"
  xmlns:lrm="urn:oma:drm:sce:lrm"
  xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance">
    <reqID>
    <keyIdentifier xsi:type="gen:X509SPKIHash">
      <hash>c2FtcGxlRFJNQWdlbnQ=</hash>
    </keyIdentifier>
  </reqID>
  <resID>
    <keyIdentifier xsi:type="gen:X509SPKIHash">
      <hash>c2FtcGxlUmlnaHRzSXNzdWVy</hash>
    </keyIdentifier>
  </resID>
  <nonce>RFJNQWdlbnROb25jZTA=</nonce>
<rspInfo>
   <lrm:LRMRIDevPubKeyAcquisitionResponseInformation>
       <DevPubKey>c2FtcGxlUmlnaHRz</DevPubKey>
   </lrm:LRMRIDevPubKeyAcquisitionResponseInformation>
  </rspInfo>
  <signature>SignatureValue</signature>
</lrm:DevPubKeyAcquisitionResponse>
```

D.9 LRM-RICreateDeviceRORequest

```
<?xml version="1.0" encoding="UTF-8"?>
<lrm:LRMRICreateDeviceRO
xmlns:ds="http://www.w3.org/2000/09/xmldsig#"
  xmlns:gen="urn:oma:drm:sce:gen"
  xmlns:xenc="http://www.w3.org/2001/04/xmlenc#"
  xmlns:lrm="urn:oma:xml:sce:lrm"
```

```
 xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance">
 <reqID>
  <keyIdentifier xsi:type="gen:X509SPKIHash">
   <hash>c2FtcGxlRFJNQWdlbnQ=</hash>
  </keyIdentifier>
 </reqID>
 <resID>
  <keyIdentifier xsi:type="gen:X509SPKIHash">
   <hash>c2FtcGxlUmlnaHRzSXNzdWVy</hash>
  </keyIdentifier>
 </resID>
 <nonce>RFJNQWdlbnROb25jZTA=</nonce>
 <time>2010-12-31T12:00:00</time>
<reqInfo>
   <lrm:LRMRICreateDeviceRORequestInformation>
     <sourceLRMID>n8yu98hy0e2109eu09ewf09u</sourceLRMID>
     <recipientDeviceID>X4yu56hy0e7709eu00Pwf0l9 </recipientDeviceID>
     <rights o-ex:id="REL1">
       <o-ex:context>
         <o-dd:version>2.1</o-dd:version>
         <o-dd:uid>n8yu98hy0e2109eu09ewf09u</o-dd:uid>
       </o-ex:context>
       <o-ex:agreement>
         <o-ex:asset>
           <o-ex:context>
             <o-dd:uid>ContentID</o-dd:uid>
           </o-ex:context>
           <o-ex:digest>
             <ds:DigestMethod Algorithm="http://www.w3.org/2000/09/xmldsig#sha1"/>
             <ds:DigestValue>bLLLc+Um/5/NvmYKiHDLaErK0fk=</ds:DigestValue>
           </o-ex:digest>
           <ds:KeyInfo>
             <xenc:EncryptedKey>
               <xenc:EncryptionMethod Algorithm="http://www.w3.org/2001/04/xmlenc#kw-aes128"/>
               <xenc:CipherData>
                 <xenc:CipherValue>EncryptedCEK</xenc:CipherValue>
               </xenc:CipherData>
             </xenc:EncryptedKey>
           </ds:KeyInfo>
         </o-ex:asset>
         <o-ex:permission>
           <o-dd:play>
             <o-ex:constraint>
               <o-dd:count>1</o-dd:count>
             </o-ex:constraint>
           </o-dd:play>
         </o-ex:permission>
       </o-ex:agreement>
     </rights>
    <encKeyInfo>
      <encKey>
        <xenc:EncryptionMethod
   Algorithm="http://www.rsasecurity.com/rsalabs/pkcs/schemas/pkcs-1#rsaes-kem-kdf2-kw-aes128"/>
        <ds:KeyInfo>
          <roap:X509SPKIHash>
            <hash>vXENc+Um/9/NvmYKiHDLaErK0gk=</hash>
          </roap:X509SPKIHash>
```

```
            </ds:KeyInfo>
            <xenc:CipherData>
                                    <xenc:CipherValue>CipherValue</xenc:CipherValue>
            </xenc:CipherData>
          </encKey>
        </encKeyInfo>
        <mac>MacValue</mac>
      </lrm:LRMRICreateDeviceRORequestInformation>
    </reqInfo>
  <signature>SignatureValue</signature>
</lrm:LRMRICreateDeviceRORequest>
```

D.10 LRM-RICreateDeviceROResponse

```
<?xml version="1.0" encoding="UTF-8"?>
<lrm:LRMRICreateDeviceROResponse
  status="Success"
  xmlns:ds="http://www.w3.org/2000/09/xmldsig#"
  xmlns:gen="urn:oma:xml:sce:gen"
  xmlns:xenc="http://www.w3.org/2001/04/xmlenc#"
  xmlns:lrm="urn:oma:drm:sce:lrm"
  xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance">
  <reqID>
    <keyIdentifier xsi:type="gen:X509SPKIHash">
      <hash>c2FtcGxlRFJNQWdlbnQ=</hash>
    </keyIdentifier>
  </reqID>
  <resID>
    <keyIdentifier xsi:type="gen:X509SPKIHash">
      <hash>c2FtcGxlUmlnaHRzSXNzdWVy</hash>
    </keyIdentifier>
  </resID>
  <nonce>RFJNQWdlbnROb25jZTA=</nonce>
<rspInfo>
    <lrm:LRMRICreateDeviceROResponseInformation>
        <success/>
      </lrm:LRMRICreateDeviceROResponseInformation>
  </rspInfo>
  <signature>SignatureValue</signature>
</lrm:LRMRICreateDeviceROResponse>
```

D.11 LRM-RICreateDomainRORequest

```
<?xml version="1.0" encoding="UTF-8"?>
<lrm:LRMRICreateDomainRO
xmlns:ds="http://www.w3.org/2000/09/xmldsig#"
  xmlns:gen="urn:oma:xml:sce:gen"
  xmlns:xenc="http://www.w3.org/2001/04/xmlenc#"
  xmlns:lrm="urn:oma:drm:sce:lrm"
  xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance">
  <reqID>
    <keyIdentifier xsi:type="gen:X509SPKIHash">
      <hash>c2FtcGxlRFJNQWdlbnQ=</hash>
    </keyIdentifier>
  </reqID>
  <resID>
    <keyIdentifier xsi:type="gen:X509SPKIHash">
      <hash>c2FtcGxlUmlnaHRzSXNzdWVy</hash>
```

```
    </keyIdentifier>
  </resID>
  <nonce>RFJNQWdlbnROb25jZTA=</nonce>
  <time>2010-12-31T12:00:00</time>
<reqInfo>
    <lrm:LRMRICreateDomainRORequestInformation>
      <sourceLRMID>n8yu98hy0e2109eu09ewf09u</sourceLRMID>
      <domainID>Domain-XYZ-001</domainID>
      <rights o-ex:id="REL1">
        <o-ex:context>
          <o-dd:version>2.1</o-dd:version>
          <o-dd:uid>n8yu98hy0e2109eu09ewf09u</o-dd:uid>
        </o-ex:context>
        <o-ex:agreement>
          <o-ex:asset>
          <o-ex:context>
            <o-dd:uid>ContentID</o-dd:uid>
          </o-ex:context>
          <o-ex:digest>
            <ds:DigestMethod Algorithm="http://www.w3.org/2000/09/xmldsig#sha1"/>
            <ds:DigestValue>bLLLc+Um/5/NvmYKiHDLaErK0fk=</ds:DigestValue>
          </o-ex:digest>
          <ds:KeyInfo>
            <xenc:EncryptedKey>
              <xenc:EncryptionMethod Algorithm="http://www.w3.org/2001/04/xmlenc#kw-aes128"/>
              <xenc:CipherData>
                <xenc:CipherValue>EncryptedCEK</xenc:CipherValue>
              </xenc:CipherData>
            </xenc:EncryptedKey>
          </ds:KeyInfo>
          </o-ex:asset>
          <o-ex:permission>
            <o-dd:play/>
          </o-ex:permission>
        </o-ex:agreement>
      </rights>
      <enc_REKs_Kmac>
        <encKey>
          <xenc:EncryptionMethod
      Algorithm="http://www.w3.org/2001/04/xmlenc#kw-aes128"/>
          <ds:KeyInfo>
            <roap:domainID>Domain-XYZ-001</roap:domainID>
          </ds:KeyInfo>
          <xenc:CipherData>
            <xenc:CipherValue>CipherValue</xenc:CipherValue>
          </xenc:CipherData>
        </encKey>
      </enc_REKs_Kmac>
      <mac>MacValue</mac>
    </lrm:LRMRICreateDomainRORequestInformation>
  </reqInfo>
  <signature>SignatureValue</signature>
</lrm:LRMRICreateDomainRORequest>
```

D.12 LRM-RICreateDomainROResponse

```
<?xml version="1.0" encoding="UTF-8"?>
```

```
<lrm:LRMRICreateDomainROResponse
 status="Success"
 xmlns:ds="http://www.w3.org/2000/09/xmldsig#"
 xmlns:gen="urn:oma:xml:sce:gen"
 xmlns:xenc="http://www.w3.org/2001/04/xmlenc#"
 xmlns:lrm="urn:oma:drm:sce:lrm"
 xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance">
  <reqID>
  <keyIdentifier xsi:type="gen:X509SPKIHash">
   <hash>c2FtcGxlRFJNQWdlbnQ=</hash>
  </keyIdentifier>
 </reqID>
 <resID>
  <keyIdentifier xsi:type="gen:X509SPKIHash">
   <hash>c2FtcGxlUmlnaHRzSXNzdWVy</hash>
  </keyIdentifier>
 </resID>
 <nonce>RFJNQWdlbnROb25jZTA=</nonce>
<rspInfo>
 <lrm:LRMRICreateDeviceROResponseInformation>
      <domainRO>
                <roap:protectedRO
                xmlns:roap="urn:oma:bac:dldrm:roap-1.0"
                xmlns:ds="http://www.w3.org/2000/09/xmldsig#"
                 xmlns:xenc="http://www.w3.org/2001/04/xmlenc#"
                xmlns:o-ex="http://odrl.net/1.1/ODRL-EX"
                xmlns:o-dd="http://odrl.net/1.1/ODRL-DD"
                 xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance">
                 <roap:ro id="n8yu98hy0e2109eu09ewf09u" domainRO="true" version="1.1"
                riURL="http://www.ROs-r-us.com">
                  <riID>
                   <keyIdentifier xsi:type="roap:X509SPKIHash">
                    <hash>aXENc+Um/9/NvmYKiHDLaErK0fk=</hash>
                   </keyIdentifier>
                  </riID>
                  <rights o-ex:id="REL1">
            <o-ex:context>
              <o-dd:version>2.1</o-dd:version>
              <o-dd:uid>n8yu98hy0e2109eu09ewf09u</o-dd:uid>
            </o-ex:context>
            <o-ex:agreement>
             <o-ex:asset>
              <o-ex:context>
                <o-dd:uid>ContentID</o-dd:uid>
              </o-ex:context>
              <o-ex:digest>
               <ds:DigestMethod Algorithm="http://www.w3.org/2000/09/xmldsig#sha1"/>
               <ds:DigestValue>bLLLc+Um/5/NvmYKiHDLaErK0fk=</ds:DigestValue>
              </o-ex:digest>
              <ds:KeyInfo>
                <xenc:EncryptedKey>
                 <xenc:EncryptionMethod Algorithm="http://www.w3.org/2001/04/xmlenc#kw-aes128"/>
                 <ds:KeyInfo>
                  <ds:RetrievalMethod URI="#K_MAC_and_K_REK"/>
                 </ds:KeyInfo>
                  <xenc:CipherData>
                   <xenc:CipherValue>EncryptedCEK</xenc:CipherValue>
```

```
                    </xenc:CipherData>
                  </xenc:EncryptedKey>
                </ds:KeyInfo>
              </o-ex:asset>
              <o-ex:permission>
                <o-dd:play/>
              </o-ex:permission>
            </o-ex:agreement>
          </rights>
          <signature>
            <ds:SignedInfo>
              <ds:CanonicalizationMethod Algorithm="http://www.w3.org/2001/10/xml-exc-c14n#"/>
                  <ds:SignatureMethod
        Algorithm="http://www. rsasecurity.com/rsalabs/pkcs/schemas/pkcs-1#rsa-pss-default"/>
              <ds:Reference URI="#REL1">
                <ds:Transforms>
                  <ds:Transform Algorithm="http://www.w3.org/2001/10/xml-exc-c14n#"/>
                </ds:Transforms>
                <ds:DigestMethod Algorithm="http://www.w3.org/2000/09/xmldsig#sha1"/>
                <ds:DigestValue>slo5hb+id8JtuOMNKs12=drf5+3df=</ds:DigestValue>
              </ds:Reference>
            </ds:SignedInfo>
            <ds:SignatureValue>j6lwx3rvEPO0vKtMup4NbeVu8nk=</ds:SignatureValue>
            <ds:KeyInfo>
              <roap:X509SPKIHash>
                <hash>aXENc+Um/9/NvmYKiHDLaErK0fk=</hash>
              </roap:X509SPKIHash>
            </ds:KeyInfo>
          </signature>
          <encKeyInfo>
            <encKey Id="K_MAC_and_K_REK">
              <xenc:EncryptionMethod
          Algorithm="http://www.w3.org/2001/04/xmlenc#kw-aes128"/>
              <ds:KeyInfo>
                <roap:domainID>Domain-XYZ-001</roap:domainID>
              </ds:KeyInfo>
              <xenc:CipherData>
                <xenc:CipherValue>32fdsorew9ufdsoi09ufdskrew9urew0uderty5346wq</xenc:CipherValue>
              </xenc:CipherData>
            </encKey>
          </encKeyInfo>
        </roap:ro>
        <mac>
          <ds:SignedInfo>
            <ds:CanonicalizationMethod Algorithm="http://www.w3.org/2001/10/xml-exc-c14n#"/>
            <ds:SignatureMethod
          Algorithm="http://www.w3.org/2000/09/xmldsig#hmac-sha1"/>
              <ds:Reference URI="#n8yu98hy0e2109eu09ewf09u">
                <ds:Transforms>
                  <ds:Transform Algorithm=http://www.w3.org/2001/10/xml-exc-c14n#/>
                </ds:Transforms>
                <ds:DigestMethod Algorithm="http://www.w3.org/2000/09/xmldsig#sha1"/>
                  <ds:DigestValue>slo5hb+id8JtuOMNKs12=drf5+3df=</ds:DigestValue>
              </ds:Reference>
            </ds:SignedInfo>
            <ds:SignatureValue>j6lwx3rvEPO0vKtMup4NbeVu8nk=</ds:SignatureValue>
            <ds:KeyInfo>
```

```
                <ds:RetrievalMethod URI="#K_MAC_and_K_REK"/>
              </ds:KeyInfo>
            </mac>
           </roap:protectedRO>
         </domainRO>
    </lrm:LRMRICreateDeviceROResponseInformation>
  </rspInfo>
  <signature>SignatureValue</signature>
</lrm:LRMRICreateDomainROResponse>
```