Introduction of the Domain Issuer in OMA DRM

Paul Koster
Philips Research Europe
High Tech Campus 34 (WB6)
5656 AE Eindhoven, The Netherlands
Phone +31 40 27 47837, Fax +31 40 27 46321
r.p.koster@philips.com

Javier Montaner
Vodafone Group R&D .NL
P.O.Box 1500
6201 BM Maastricht, The Netherlands
javier.montaner@vodafone.com

Sorin Iacob
Telematica Instituut
P.O.Box 589
7500 AN Enschede, The Netherlands
sorin.iacob@telin.nl

Najib Koraichi
Vodafone Group R&D .NL
P.O.Box 1500
6201 BM Maastricht, The Netherlands
najib.koraichi@vodafone.com

Abstract: We present the introduction of the Domain Issuer in the Open Mobile Alliance digital rights management (OMA DRM) standard. Domains enable users to access content on multiple devices. In OMA DRM 2.0 Rights Issuers manage which devices belong to a domain besides issuing licenses. However, when users buy content from different providers and must maintain a domain at each of them this results in a confusing customer experience. Sharing domains over multiple content providers solves this problem. Therefore, this paper presents an architecture supporting the introduction of the Domain Issuer that manages this shared domain.

Key words: DRM, OMA DRM, DRM architecture, domains, DRM protocols

Introduction of the Domain Issuer in OMA DRM

Paul Koster
Philips Research Europe
High Tech Campus 34 (WB6)
5656 AE Eindhoven, The Netherlands
r.p.koster@philips.com

Javier Montaner, Najib Koraichi
Vodafone Group R&D .NL
P.O.Box 1500
6201 BM Maastricht, The Netherlands
{javier.montaner|najib.koraichi}
@vodafone.com

Sorin Iacob
Telematica Instituut
P.O.Box 589
7500 AN Enschede, The Netherlands
sorin.iacob@telin.nl

***Abstract*—We present the introduction of the Domain Issuer in the Open Mobile Alliance digital rights management (OMA DRM) standard. Domains enable users to access content on multiple devices. In OMA DRM 2.0 Rights Issuers manage which devices belong to a domain besides issuing licenses. However, when users buy content from different providers and must maintain a domain at each of them this results in a confusing customer experience. Sharing domains over multiple content providers solves this problem. Therefore, this paper presents an architecture supporting the introduction of the Domain Issuer that manages this shared domain.**

Keywords: DRM, OMA DRM, DRM architecture, domains

I. INTRODUCTION

The OMA DRM 2.0 standard is recently finalized [1,2]. It features a domain concept to enhance the user's DRM experience. This domain concept enables a user to register upto a maximum number of devices on which he can share and access his content. In OMA DRM 2.0 users hold a domain per content provider. OMA DRM 2.0 provides domain functionality through the Rights Issuer (RI) functionality. A user registers all his devices to each RI belonging to the content providers from which he buys content. This results in consumer confusion if some of his OMA DRM content does play on one of his devices while other content does not and first requires online registration of the device. This problem gets worse when a user has registered the maximum number of devices at individual content providers, but those devices are different per content provider. The latter prevents manual synchronization of domains.

We present an extension to the OMA DRM 2.0 architecture that enables users to share the same domain over multiple content providers. Key aspect is the introduction of a Domain Issuer (DI) next to the Rights Issuer (RI). Contributions of this paper include the architecture, key management and protocols. The solution adheres to the principles underlying OMA DRM.

The structure of this paper is as follows. Section II briefly presents the current OMA DRM 2.0 architecture. Section III characterizes the solution and presents the main requirements. Section IV discusses the architecture and key management. Section V evaluates the solution against the requirements. and section VI presents a security analysis of the solution. The paper ends with conclusions in section VII.

This work was sponsored by the MESEC project.

II. OMA DRM 2.0

OMA DRM 2.0 [1,2] is a digital rights management system to protect premium multimedia content. Its main concepts, depicted in Figure 1, are Right Objects (RO) representing licenses, Rights Issuers (RI) responsible for license issuing via the Rights Object Acquisition Protocol (ROAP), and DRM Agents (DA) responsible for DRM client functionality including license evaluation and content decryption. ROs comprise usage rules and protected content keys. Before a RI and a DA interact they perform the RI Registration protocol to exchange information and cryptographic keys, and to verify each others compliance to the OMA DRM system. This results in a RI Context at the DA. To manage compliance DAs and RIs are certified. RIs have access to certificate revocation lists, and to OCSP (Online Certificate Status Protocol) responders to make up to date revocation information about RIs available to DAs. Public/private key pairs associated with the certificates are used to sign and encrypt protocol messages.

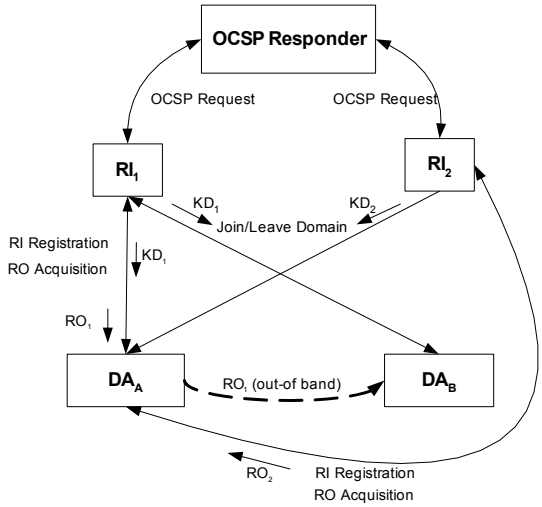

Figure 1. OMA DRM 2.0 architecture. DA_A obtains licenses from RI_1 and RI_2 and can use both. DA_B obtains a license issued by RI_1 from DA_A via an out of band protoocol and can use it.

OMA DRM 2.0 supports domains to enable a user to access his content on multiple DAs. RIs manage domains according to a domain policy. Key management is based on a symmetric

domain key (KD), which protects the content key contained in the RO. The domain key is distributed securely to DAs via the Join Domain protocol. Successful completion of this protocol results in a Domain Context at the DA. RIs can update the domain key to a new generation when they consider this necessary for security purposes. To improve performance OMA DRM supports key hash chains which enable a DA to derive previous generations of the domain key from the current domain key. To exchange ROs between DAs any out of band mechanism may be used, i.e. not using the ROAP protocol. For example, ROs may be embedded in the protected DCF (DRM Content Format) content container.

III. THE DOMAIN ISSUER

As argued above a functional separation of license management and domain management in OMA DRM is desired. RI stays responsible for license management, and the Domain Issuer becomes responsible for domain management. Domain management independent from license management is known [3,4]. However, these approaches do not position the domain manager as an online service. Furthermore, these approaches hardly address the (security) interests of the domain manager itself.

A DI should be able to run as an independent service, which means that it must also be able to compensate the costs involved with running the service. The architecture should enable the DI to engage in relationships with both end-users and RIs. The main requirements for the introduction of the DI are:

R1. The DI should play an essential role in key management for his domains so that it cannot be bypassed.

R2. The DI should be able to stop the use of domain functionality when the business relationship with RI ends.

R3. The RI shall trust the DI but should not need to trust the other RI that issue content for the same domain.

R4. Non-trusted devices should be revocable from a domain in order to secure future domain content.

R5. Content issued by other RIs should not be affected when a RI is revoked.

R6. Content acquired for a domain should remain accessible after a DI ceases to exist.

R7. Runtime dependencies and protocol interactions between business entities should be minimized.

R8. User friendliness and experience should be promoted and at least be as good as OMA DRM 2.0.

IV. ARCHITECTURE AND KEY MANAGEMENT

A. Architecture

The extended OMA DRM architecture to introduce the DI starts with a number of high impact considerations. First, is there one or more DIs that decide on the domain membership? In the latter case the DIs must synchronize the domain composition, i.e. which devices belong to the domain. From a practical perspective one DI is favored. This does not result in loss of functionality, is clearer from a control perspective, and is less confusing for users than multiple DIs. Therefore, a domain is associated to a single DI that manages it. This domain can be used by several RIs.

Second, is key management based on one domain key per DI, one domain key per RI, or a completely different scheme based on public/private keys? Consequently, this also determines who defines keys in the key hierarchy. A domain key per DI is selected. This gives the DI control over its domains, is practical since there are less keys to manage, and is user friendly since DAs can obtain the required keys via one online transaction. A key per RI would severely impact distribution of (updated) domain keys. A public/private key pair as domain key defined by DI with the public keys distributed to the RIs has the disadvantage that there is no efficient update mechanism like key hash chains available, although it has the advantage that it complies with requirement R3.

Third, what communication pattern between RI, DI and DA is followed? Basic business principles and simplicity indicate that DA needs to contact directly with an RI to obtain ROs, and also directly with a DI to join/leave a domain. Communication of domain information, e.g. the domain composition, between DI and RI is required for a RI to issue ROs for a domain. This communication can be done directly or indirectly. An example of indirect communication is that DIs issue membership certificates to DAs which pass it on to RIs. However, this provides no easy mechanism to signal to RIs that a device left a domain. For reasons of robustness protocol interactions are restricted to two participating parties.

The above considerations are input to the general architecture. This architecture is summarized below and illustrated in Figure 2. The following sections discuss each item in more detail.

- Master Domain Key (KD) shared by DI and DAs and not available to RIs.
- Diversified Domain Keys (KD_i) per RI_i.
- Validation Tokens (VT) entitle RIs to issue content to domains of a DI.
- Registration protocol between DA and DI establishes trust between DI and DA.
- Join/Leave Domain protocols between DA and DI replace the protocols between DA and RI.
- Use Domain protocol between RI and DI distributes KD_i and VT.
- RO Acquisition protocol also takes KD_i and VT into account.
- DAs append a DeviceMAC to acquired ROs to allow out-of-band RO distribution.

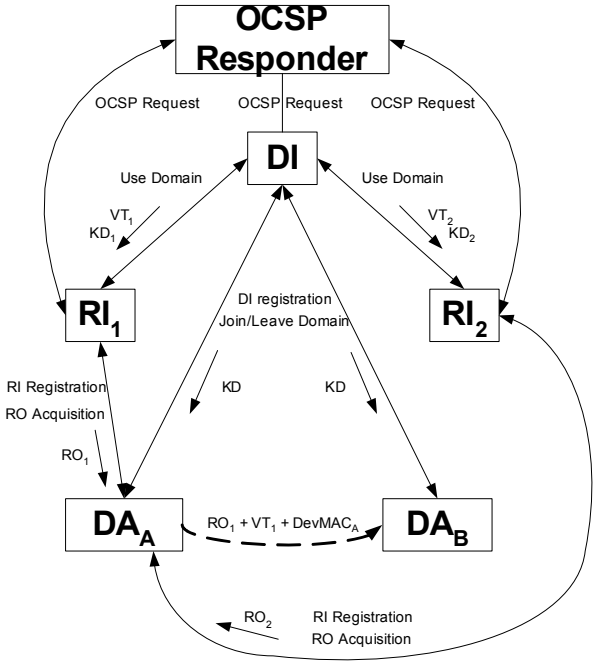

Figure 2. Enhanced architecture and key management for the introduction of the domain issuer in OMA DRM.

B. Key Management

Key management involves three new concepts:

1) Domain Key Diversification: domain key diversification gives RIs independent keys (KD_i) to meet requirement R3. The DI and DA calculate KD_i using the diversification function:

KD_i = first 128 bits of HMAC-SHA1(PubKeyRI, KD)

Instead of the public key of the RI any public information can be used, but the advantage of the public key is that it is already available to the DAs. Note that cryptographic primitives already present in the OMA DRM specification [2] are reused.

2) ValidationToken: VTs offer a DI control over which RIs can issue ROs to a domain, i.e. requirement R1 and R2. A DI generates a VT and distributes it to a RI using the Use Domain protocol. A VT is renewed periodically. The RI uses the VT to prove towards a DA that it may issue ROs for the domains of this DI. A VT is defined like a certificate or assertion:

VT_i = {ExpiresAfter, RI_i PublicKey}signedDIPrivateKey

3) DeviceMAC: DeviceMAC (Message Authentication Code) complements VT. It targets the same requirements, but covers the out of band exchange of ROs between DAs instead of RO acquisition from the RI. A DeviceMAC serves as a proof by a DA that the RI at the moment of the RO acquisition protocol had a valid non-expired VT. This prevents a RI to create and distribute ROs out of band after its relation with the DI has ended. A DA computes a MAC of the RO using KD after acquiring a RO. DAs must validate DeviceMAC before installing out of band received ROs. DAs calculate DeviceMAC over (essential parts of) a RO:

DeviceMAC = HMAC-SHA1(RO, KD)

C. Protocols

The main protocols are discussed below. Figure 3 depicts the individual protocol messages for a common scenario.

1) DI Registration: The DI Registration protocol comprises an exchange of security information between the DI and the DA. The DA initiates the protocol at first contact. Optionally, the protocol is run later to update the information. The protocol message interaction follows the RI Registration protocol of OMA DRM 2.0 except that a DI instead of a RI is contacted.

Successful completion of the protocol results in the establishment of a DI Context at the DA. The DI Context includes agreed protocol parameters, protocol version, certificates, etc. A DI Context is necessary for the other protocols including Join/Leave Domain and RO Acquisition.

2) RI Registration: The RI Registration protocol comprises an exchange of security information between the RI and the DA. For example, the registration includes exchange of certificates and validation thereof. It is used as defined in OMA DRM 2.0.

3) Join/Leave Domain: The Join/Leave Domain protocol between DA and DI is the protocol by which a DA joins/leaves a domain. The protocols assume an existing DI Context. It is similar to the Join/Leave Domain protocols of OMA DRM 2.0. The difference is that a DA contacts a DI instead of a RI.

A DI permits or denies the Join Domain Request depending on the domain policy. Successful completion of the Join Domain protocol results in the establishment of a Domain Context at the DA. This context contains domain specific security parameters including the Domain Key (KD). The DA uses KD to calculate KD_i per RI.

Successful completion of the Leave Domain protocol results in deregistration of the DA at the DI and deletion of the Domain Context from the DA.

To deregister a DA from the domain a DI may send a Leave Domain Trigger to the DA. Furthermore, a DI may issue a new generation of KD such that a DA cannot access any new content as in OMA DRM 2.0. It informs the RIs of the updated KD_i by sending them Use Domain Triggers.

4) Use Domain: The Use Domain protocol enables a RI to use a domain managed by a DI. A contractual agreement between RI and DI is assumed providing the required trust and other arrangements, e.g. the domain policy. Use Domain is a new protocol supporting the logical split of DI and RI functionality.

Successful completion of the Use Domain protocol results in an (updated) Use Domain Context (UDC) at the RI containing domain specific security related information. A Use Domain Context is necessary for the RO Acquisition protocol. It is defined as:

UDC = DI Id, Domain Id, KD_i, ExpiresAfter, VT_i

RI identifies the domain and the associated DI in the Use Domain Request message. DI creates KD_i for this domain and distributes it to the RI as part of the Use Domain Response message. The RI stores the latest version of this diversified domain key.

DI also generates the RI validation token and distributes this to the RI within the Use Domain Response message. This token is renewed periodically by the DI. An update may be requested by RI when VT is about to expire, or DI sends a Use Domain Trigger to initiate a VT renewal. Before issuing a VT DI checks that the RI is still compliant and has an agreement with the DI.

Disallowing a RI to use the domain is achieved by stopping the distribution of new validation tokens VT. In the case that the DI does not trust a RI, the DI can also create a new generation of KD and distribute it to all the DAs in the domain and issue new KD_i to all the RIs except the non-trusted RI.

5) Rights Object Acquisition: The RO Acquisition protocol enables a DA to acquire ROs. The DA must have a pre-established RI Context and DI context. The RI must have a pre-established Use Domain Context with DI. The RO Request message defined in OMA DRM 2.0 is extended such that it identifies the DI similarly as it identifies the RI.

RI creates the RO for the given domain as it does in OMA DRM 2.0. New is that it protects the RO using KD_i instead of KD. It also adds the DI identity to the RO. Furthermore, RI includes the VT as a new XML element in the RO. It distributes the RO in the RO Response message.

When DA receives the RO Response message it needs to validate it. This includes the checks already defined by OMA DRM 2.0 such as verification of the RI signature. Furthermore, DA checks the expiry date in the VT and decides whether the installation of the RO can proceed or the RO must be discarded. Also, devices may only accept ROs protected by the latest generation of the domain key. If the generation used by the RI is older, the RO is rejected. If the generation used by the RI is newer, the device should run the Join Domain protocol to obtain the new domain key before proceeding with the acceptance of the RO.

RIs must distribute ROs using the RO Acquisition protocol to a DA that is a member of the domain. To enforce this DAs perform above checks and compute a DeviceMAC after accepting the RO. The DA adds the DeviceMAC as a new XML element to the RO, which is subsequently stored. This element is added such that it does not affect the part of the RO that is integrity protected.

6) Out of band RO exchange: Other then through the RO Acquisition protocol, devices can also receive domain ROs by different means. An example is the case where the RO is embedded in the secure content container and subsequently copied to another device. Before using these ROs, DAs need to perform some validation checks.

DAs must have valid DI and domain contexts. The DA first reads the identity of the DI and of the domain from the RO. It then checks if valid DI and domain contexts with the valid KD generation exist. Then the device uses the DI Public Key contained in the DI/Domain context to check the signature of the VT received with the RO. It does not check the validity date of the VT. Next, it checks the RO signature with the RI Public Key value contained in the VT. Finally, it checks the DeviceMAC using KD.

The OMA DRM 2.0 requirement that DAs must have a valid RI context is dropped, because it could result in user confusion in the following scenario (requirement R8). Consider a user who has defined a domain with a number of DAs. Now he acquires a RO on one domain DA from a new RI. Then the user copies this RO to another domain DA using an out of band mechanism. When he later tries to access this RO when this DA is offline it fails because the DA does not have the RI context, while all other ROs from different RIs do play. Although the extended architecture does not require a valid RI Context, RI compliance is still assured via the DA, which performs the RO Acquisition protocol and generates DeviceMAC if the RI is compliant.

Only if all the validations are correct then the RO can be used. In case of a content access KD_i is computed from KD using the diversification function. KD_i is then used to obtain the content encryption key from the RO.

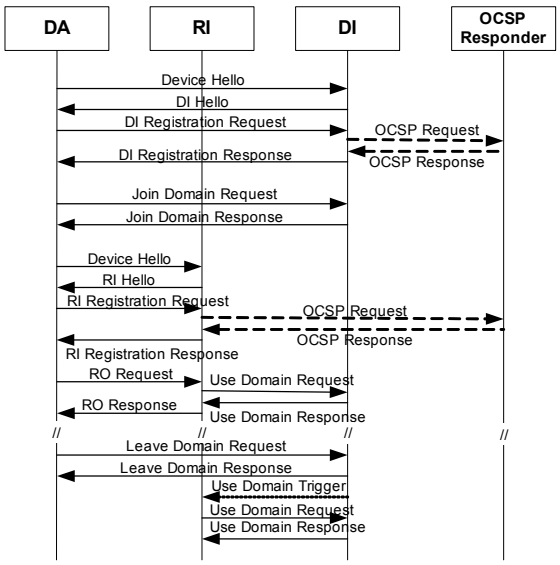

Figure 3. Interaction diagram for a basic scenario where a DA registers at a DI, joins a domain, registers at a RI, and acquires an RO at the RI for the domain. RI interacts with DI to obtain the domain key. Furthermore, DA leaves the domain, which triggers a Use Domain protocol, e.g. to update the domain key or to communicate the new domain composition.

V. EVALUATION

The solution introducing the domain issuer addresses the requirements as follows. The interests of the DI (R1 and R2) are addressed by the combination of DI issuing the domain keys, Validation Tokens and DeviceMAC.

RIs sharing a domain do not need to trust each other (R3), because the diversified domain key ensures that each RI protects its content using a key not known by other RIs.

A domain key upgrade by DI addresses R4. The new KD is not distributed to the revoked device. This way future content is secure like it is in current OMA DRM 2.0.

In case an RI is revoked content of other RIs is not affected (R5), because VT is RI specific. If it is chosen to update KD without distributing KD_i to the revoked RI then new content of the other RIs is slightly affected since a Join Domain is required by DAs to obtain the new key. Old content is not affected.

When a DI ceases to exist the content is not available over time, because new devices cannot obtain DK. Therefore, R6 is not met by the architecture. This requires a transfer procedure whereby domains are transferred from one DI to another.

The architecture minimizes runtime dependencies (R7). Regularly RIs need to contact the DI via the Use Domain protocol, but this can be done independent of other protocols. This contributes to availability and robustness as properties of the architecture.

The solution is user friendly (R8). From the user/device perspective, domain management only requires an interaction with the DI. The interaction between RIs and DI is not visible from the perspective of the end-user. Adding or removing a device from the domain does not have impact on other domain devices. Similarly, buying from a new RI is transparent since no RI Context is required anymore at a DA.

VI. Security Analysis

The introduction of the DI has impact on security. In the assumed threat model both users and service providers (RI and DI) may attack the system. Attacks and countermeasures are discussed below.

1) Compromise of device private key or device non-compliance to compliance and robustness rules. A compromised device private key can be used to obtain domain and content keys and bypass content protection. The existing OMA DRM 2.0 countermeasure, device revocation, suffices.

2) Compromise of domain key, which key can be used to obtain content keys and bypass content protection. The existing countermeasure(s) suffice: device, RI, or DI revocation depending on the source of attack, and new domain key generation.

3) Compromise of RI private key or RI non-compliance to compliance and robustness rules. This attack enables countent loundering. OMA DRM 2.0 checks if the certificate of RI is revoked at RI registration with a DA. However, the introduction of the DI makes it possible that a device does have a DI context, but not a RI context and therefore also no OCSP response asserting that the certificate of RI is not revoked. The architecture addresses this via VT and DeviceMAC. The Validity Token certifies the RI public key at time of issuing an RO to a DA and DeviceMAC forwards this to other DAs. These objects imply RI validation at the time it issues ROs.

4) Compromise of DI private key or DI non-compliance to compliance and robustness rules. The countermeasure consists of DI revocation similar to RI revocation, which is is checked by a DA at DI registration using OCSP. Furthermore, a RI may refuse to issue ROs for domains managed by the DI.

5) RI uses domain without RI-DI agreement or continues to use domain after termination of RI-DI agreement. Validation Tokens which must be renewed periodically counters this threat. DeviceMAC counters the threat where ROs are delivered out of band instead. Collusion between a RI and DA make it possible to issue ROs to a domain, but it is not sustainable due to DA revocation.

6) Lost or stolen device, which also causes loss of content. Similar to OMA DRM 2.0 this is considered out of scope.

7) RI leaks domain key(s), which damages other RIs' interests. The diversification of domain keys prevents this attack.

8) DI or DA leaks domain key(s), which damages RIs' interests because it discloses domain and content keys. Existing content is compromised, but new content can be protected by an update of the domain key.

9) Attacks on cryptographic primitives used in key management or protocols. Only proven cryptographic primitives already in use by OMA DRM 2.0 are used.

10) Key diversification and DeviceMAC are used to derive KD. The use the of the keyed hash function (HMAC) counters this.

VII. Conclusions

The introduction of the Domain Issuer in OMA DRM 2.0 allows users to manage one domain at one DI and use this domain at multiple Rights Issuers to obtain Rights Objects for content. This increases the user friendliness.

The extensions and changes to OMA DRM 2.0 are limited. One new protocol is added, while others are reused or extended. The architecture enables DI to engage in relationships both with end-users and RIs. Therefore two new mechanisms are added, Validation Tokens and DeviceMAC. RIs are separated from each other by diversifying the domain key.

Future work includes a mechanism to transfer domains from a DI to another. Furthermore, this research did not consider rights and/or domain management functions to be local to devices. This is subject of further research.

Acknowledgment

We would like to thank Koen Vrielink, Bob Hulsebosch and Joost Reuzel for their contributions to this work.

References

[1] Open Mobile Alliance, *DRM Architecture: Approved Version 2.0*, 2006

[2] Open Mobile Alliance, *DRM Specification: Approved Version 2.0*, 2006

[3] R.P. Koster, F.L.A.J. Kamperman, P.J. Lenoir, K.H.J. Vrielink, "Identity based DRM: Personal Entertainment Domain", *Conference on Communications and Multimedia Security*, Salzburg, Austria, September 19-21 2005, LNCS, vol 3677, Springer, pp 42-54, 2005

[4] Thomson, *SmartRight: Technical white paper, version 1.7*, 2003