August 2006

Distributed Management of OMA DRM Domains

Harikrishna Vasanta
QuSec, India

Reihaneh Safavi-Naini
University of Wollongong, rei@uow.edu.au

Nicholas Paul Sheppard
University of Wollongong, nps@uow.edu.au

Martin Jan Surminen
University of Wollongong, surminen@uow.edu.au

Distributed Management of OMA DRM Domains

Abstract

Version 2.0 of the Open Mobile Alliance's Digital Rights Management Specification provides for protected content to be shared amongst a collection of devices in a domain. Domains are created and managed directly by the rights issuer that issues rights to the domain. In this paper, we propose to devolve the management of domains to a domain manager known as "Heimdall" that acts as a broker between the devices in an authorised domain and any content providers from which content for the domain can be sourced. We describe and compare three different modes in which Heimdall might operate.

Keywords

digital rights management, authorised domains, OMA

Disciplines

Physical Sciences and Mathematics

Publication Details

This paper was originally published as: Vasanta, H, Safavi-Naini, R, Sheppard, NP & Surminen, JM, Distributed Management of OMA DRM Domains, 7th International Workshop on Information Security Applications 2006 (WISA 2006), Jeju Island, Korea, 28-30 August 2006, Lecture Notes in Computer Science 4298/2007, Springer-Verlag 2007, 237-251. The original publication is available here through Springerlink.

Distributed Management of OMA DRM Domains

Harikrishna Vasanta Reihaneh Safavi-Naini
Nicholas Paul Sheppard Jan Martin Surminen

Abstract

Version 2.0 of the Open Mobile Alliance's Digital Rights Management Specification provides for protected content to be shared amongst a collection of devices in a *domain*. Domains are created and managed directly by the rights issuer that issues rights to the domain. In this paper, we propose to devolve the management of domains to a domain manager known as *Heimdall* that acts as a broker between the devices in an authorised domain and any content providers from which content for the domain can be sourced. We describe and compare three different modes in which Heimdall might operate.

1 Introduction

Digital rights management (DRM) systems are used to control the use and distribution of copyrighted content. Copyright owners' fears of financial losses caused by widespread copyright infringement have seen digital rights management become a very active field of research over the past decade.

Early digital rights management systems worked by protecting content in such a way as to render it usable on only one device. In real life, a group of users who share similar interests might like to access content as a group, and individual users would like to access content using any of the devices that they own. The users would like to obtain content from multiple rights issuers.

Recognising this, numerous DRM systems have been proposed that support the concept of an *authorised domain* [1, 3, 5, 6, 9, 10]. An authorised domain is a group of devices that may share access to a pool of content that has been granted to that domain. A typical authorised domain, for example, may consist of all of the devices within one household.

The Open Mobile Alliance (OMA) is an organisation that specifies mobile service enablers that ensure service interoperability across devices, geographies, service providers, operators, and networks. Of particular interest to this paper, OMA has recently approved the enabler for OMA DRM Version 2.0 [8], which specifies a digital rights management system for use with mobile phones.

The OMA DRM system consists of

- *rights issuers* (RIs) who are responsible for (1) providing *rights objects* (ROs) that permit access to protected content and (2) managing domains within which ROs may be shared; and

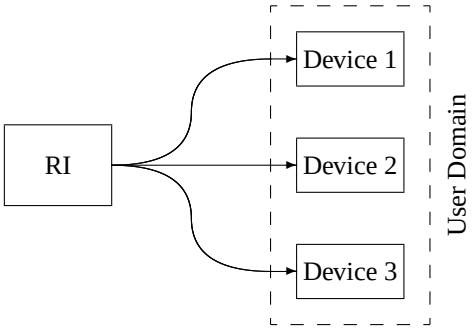

Figure 1: OMA DRM domain system.

- *DRM agents* that permit users to consume protected content according to the rights specified in ROs.

Figure 1 shows a domain containing three devices.

The base OMA specification requires that devices interact individually with every RI from which they wish to obtain content. In this paper, we propose to introduce a broker between a domain and an arbitrary number of RIs that

- relieves RIs from interacting with every domain member individually;
- provides a single sign-on point through which user devices can access all of the content to which they are entitled; and
- provides a caching service that reduces the level of traffic between RIs and user devices.

Architectures of this kind can also be used to provide inter-operability between devices and right issuers supporting a number of different digital rights management regimes [4], but in this paper we only consider the OMA DRM regime.

In order to introduce the broker, we separate the function of the RI into two components:

- the *functional responsibilities* of creating domains and providing ROs for the domain; and
- the *group management responsibilities* of admitting devices to and removing devices from the domain.

Functional responsibilities will remain with the RI but group management responsibilities will be devolved to the broker.

We call the broker *Heimdall*, after the Norse deity charged with guarding the bridge that links the realm of the gods with the realm of humans. Our Heimdall is a software application or hardware device that interacts with RIs on one hand and the user devices in a domain on the other, as shown in Figure 2.

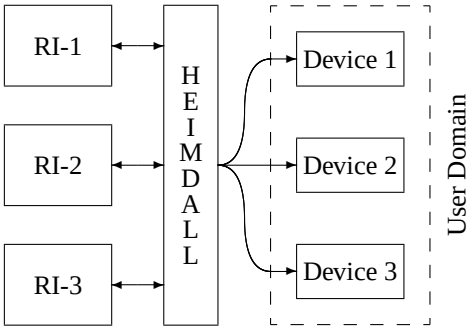

Figure 2: Our proposed domain management system.

A stand-alone instance of Heimdall may be installed at a location shared by the domain members, or Heimdall may be implemented as a service by network carriers. Heimdall is registered to all of the rights issuers that the domain members wish to use, and thereafter all communication between users' devices and the rights issuers is conducted via Heimdall.

1.1 Paper Organisation

We will give a description of the OMA DRM Version 2.0 specification in Section 2. We will then describe and analyse three different modes in which Heimdall might operate, with increasing degrees of responsibility placed on Heimdall:

- as a simple relay in Section 3;
- as a member of the domain in Section 4; or
- as a rights issuer in its own right in Section 5.

We describe how each mode can be implemented using the OMA RO Acquistion Protocol suite, and argue that the security properties of the new system are equivalent to those of the base OMA specification.

We will then give a comparison of the three modes and discuss Heimdall's relationship with other domain management frameworks in Section 6. Finally, we will conclude the paper in Section 7.

2 OMA DRM

2.1 Security Model

The security of the OMA DRM system depends on DRM agents being certified by the Content Management License Administrator (CMLA) [2] to meet tamper-resistance requirements specified by OMA. These requirements are designed to prevent dishonest

users from extracting unprotected content, decryption keys or devices' private keys from devices. Every certified device has a unique private/public key pair.

Protected content is distributed in an encrypted format called the *DRM Content Format* (DCF). Each DCF file is encrypted using a random *content encryption key* (CEK) and can be freely distributed using any convenient method. The content encryption key is included in any RO that awards permission to use the associated content, and the sensitive parts of the RO (including the CEK) are encrypted using a *rights encryption key* (REK). The REK for an RO must be obtained from the RI that issued that RO using the Rights Object Acquisition Protocol (ROAP), which will be described in detail below.

In addition to the supplying the CEK, the RO sets out what the recipient device is permitted to do with the content (play, install, etc.) and under what constraints the the content may be used (the number of times it may be played, etc.). In this paper, we are only concerned with the cryptographic components of ROs.

The integrity of an RO is protected by having it signed by the RI that issued it. A DRM agent must obtain the RI's certificate chain using ROAP messages and verify the RI's signature before using an RO. This prevents dishonest users from modifying ROs in order to grant themselves permissions that have not been granted by a recognised RI.

2.2 Domains

OMA DRM uses the concept of a *domain* to share content among a group of users. Domains are created by an RI, and DRM agents may join or leave a domain by making a request to the RI that created the domain. ROs intended for the domain are encrypted using an REK itself encrypted with a *domain key* that is unique for that domain. A DRM agent receives the domain key upon joining a domain, and deletes it after leaving a domain, using protocols described below.

2.3 ROAP Messages

OMA DRM specifies four protocols for obtaining ROs and managing domains. All of the protocols are executed between an RI and a DRM agent. Every protocol may be initiated by a DRM agent, or an RI can request that a DRM agent begin the protocol by sending it a *trigger*. A typical sequence of messages is shown in Figure 3.

Registration. Before a DRM agent can process ROs issued by some RI, it must execute the *Registration Protocol* with that RI. This protocol allows the RI and DRM agent to exchange parameters; the RI to verify that the DRM agent has been certified by the CMLA; and the device to request that the RI prove that its certificate chain is still valid using the Online Certificate Status Protocol (OCSP) [7].

Join Domain. The Join Domain Protocol is used to join a device to a domain after it has been registered to the RI that controls that domain. After a successful run of the protocol, the client will have the domain key.

Rights Object Acquisiton. The Rights Object Acquisition Protocol enables DRM agents to obtain rights objects from an RI for content that has been protected by that RI. After successfully completing the protocol, the DRM agent has the RO required to use the content and the REK for that RO.

Leave Domain. A DRM agent can leave a domain by deleting all of the information associated with that domain (including the domain key) and initiating the Leave Domain Protocol with the RI.

In the Heimdall framework, Join Domain, RO Acquisition and Leave Domain requests will be processed by Heimdall as a proxy for RIs. In the mode described by Section 5, Heimdall will also process registration requests.

2.4 Security Goals

The goal of an attacker in a DRM system is to exercise rights over content that have not been granted by a legitimate rights issuer, that is, to perform an action that is not permitted by any valid rights object.

The OMA DRM specification analyses various attacks that can be mounted on the DRM system. The specification assumes that an adversary is able to:

- listen to the communication channels between the DRM agent and RI; and
- read, modify, remove, generate and inject messages in this channel.

We will require our Heimdall framework to meet the same requirements for defeating these attacks as the base OMA system.

3 Option 1: Heimdall as a Relay

In the first approach, Heimdall simply acts as a relay. Heimdall does not have access to the domain key or the decrypted ROs at any time, and therefore does not need to meet high tamper-resistance requirements. RIs are responsible for creating domains and providing ROs to devices in the domain, as well as creating the Join Domain and Leave Domain responses for each device in a domain. An example sequence of messages is shown in Figure 8.

3.0.1 Initialisation.

Devices use the Registration Protocol to register directly with any RI they wish to use prior to the initialisation of Heimdall. Users wishing to form a domain must communicate the proposed members of the domain to the RI, and supply the identity of the instance of Heimdall that they wish to use for managing the domain. If the domain is successfully created, the RI provides Heimdall with the response to a Join Domain request for each of the devices in the proposed domain, as if that device had requested to join the domain. These responses will later be forwarded to devices when they transmit a Join Domain request to Heimdall. Note that Heimdall must be initialised separately

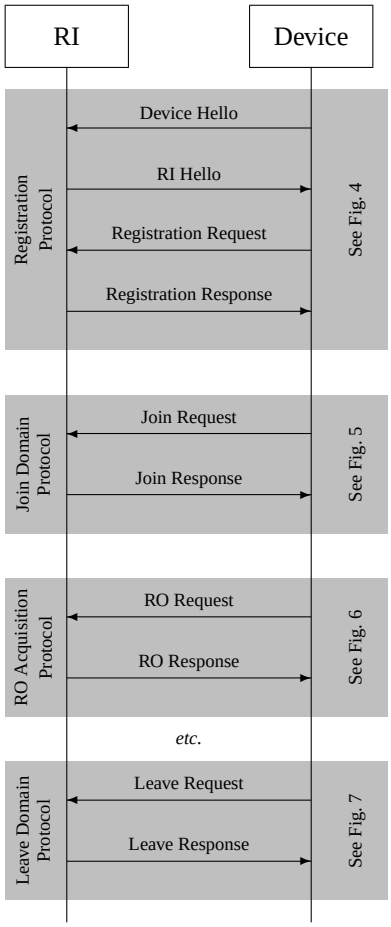

Figure 3: A typical sequence of messages in an OMA domain. Each shaded box denotes a protocol defined by OMA.

Device Hello:	ROAP Version, Device ID, Supported Algorithms, Extensions
RI Hello:	Status, Session ID, ROAP Version, RI ID, Selected Algorithms RI Nonce, Authorities, Server Info., Extensions
Registration Request:	Session ID, Device Nonce, Request Time, Cert. Chain, Authorities, Server Info., Extensions, Signature
Registration Response:	Status, Session ID, RI URL, Cert. Chain, OCSP Resp., Extensions, Signature

Figure 4: OMA Registration Protocol. The signatures in the request and response messages are over all data transmitted so far in the protocol.

Join Domain Request:	Device ID, RI ID, Device Nonce, Request Time, Domain ID, Cert. Chain, Extensions, Signature
Join Domain Response:	Status, Device ID, RI ID, Device Nonce, Domain Info., Cert. Chain, OCSP Resp., Extensions, Signature

Figure 5: OMA Join Domain Protocol.

RO Request:	Device ID, Domain ID, RI ID, Device Nonce, Request Time, RO Info, Cert. Chain, Extensions, Signature
RO Response:	Status, Device ID, RI ID, Device Nonce, Protected ROs, Cert. Chain, OCSP Resp., Extensions, Signature

Figure 6: OMA RO Acquisition Protocol.

Leave Domain Request:	Device ID, RI ID, Device Nonce, Request Time, Domain ID, Cert. Chain, Extensions, Signature
Leave Domain Response:	Status, Device Nonce, Domain ID, Extensions

Figure 7: OMA Leave Domain Protocol.

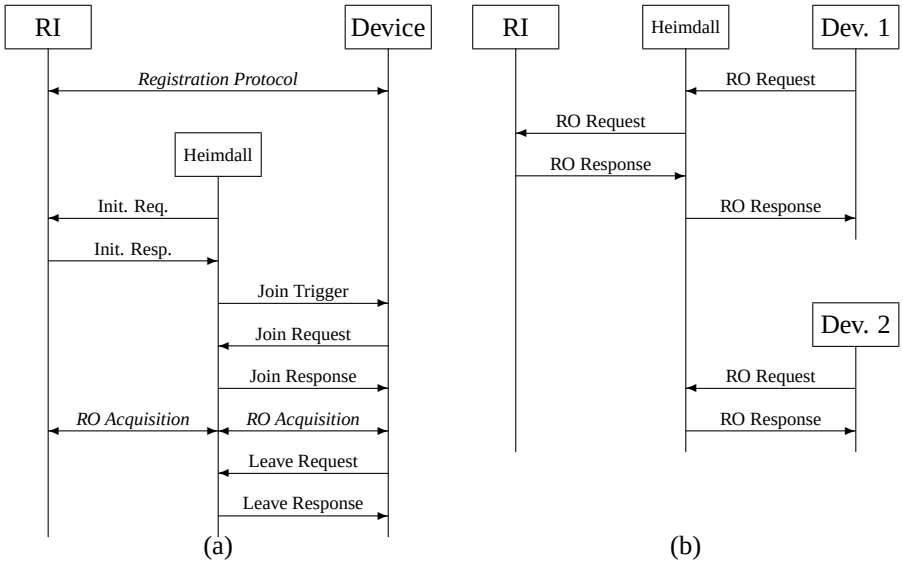

Figure 8: (a) Creating, joining and leaving a domain in Option 1 and (b) acquiring a rights object.

for every domain that it manages in order to obtain the appropriate Join Domain responses for that domain, even if those domains are created by the same RI.

3.0.2 Joining Domains.

After Heimdall has been initialised for a domain, it sends a Join Domain Trigger to all of the devices enrolled with the RI to be in this domain. The devices then join the domain by completing the Join Domain Protocol with Heimdall, which uses the pre-prepared responses that it received from the RI. Completion of this protocol gives the devices access to the domain key.

3.0.3 RO Acquisition.

Once devices have joined the domain, they can request ROs for the domain from Heimdall. If Heimdall has the desired RO, it provides this RO to the device. Otherwise, Heimdall forwards the request to the RI and obtains the RO Response. It forwards the RO Response to the requesting device, and stores a copy itself in order to serve any future requests for the same RO.

3.0.4 Leaving Domains.

Devices can leave a domain by executing the Leave Domain Protocol with Heimdall.

3.1 Security

Device registration in this mode is identical to that used in the base OMA specification and so obviously has the same security properties.

The OMA specification requires nonces to be used in request/response pairs in order to prove that the RI is "live", that is, responses are being computed in response to a particular request and not being replayed from storage. Caching the Join Domain and RO Responses on Heimdall obviously breaks this requirement, and standard OMA devices will reject forwarded Join Domain and RO Response messages because their nonces do not match. In the remainder of this section, we will assume the use of non-standard devices that do not respect the nonce but otherwise behave as normal OMA devices.

Since devices have been pre-approved for joining the domain during the initialisation phase, replaying the Join Domain Response will not gain an attacker any privileges that he or she didn't have already. If Heimdall or an RI wishes to begin refusing permission for a device to enter a domain, however, a *domain upgrade* must be performed. This procedure is defined by OMA and allows a compromised domain key to be renewed by requiring legitimate devices to re-join the domain in order to receive a new domain key. In the present context, this means re-initialising Heimdall.

OMA provides protection against replay attacks on ROs separately from the acquisition protocol. ROs containing usage constraints that require state information (such as a counter or meter) to be kept must have globally unique identifiers and devices must securely store the identifiers of any such ROs that they have been given. If a purportedly new RO arrives with the same identifier as one that has already been seen by the device, the new RO must be rejected. ROs without stateful constraints can be replayed without any affect on the security of the system. Any replay attack on the RO Acquisition Protocol, therefore, will be caught by the devices' replay protection system.

Aside from the foregoing observation about liveness, it is is easy to see that Heimdall in this mode simply forms a channel between the RI and devices that is no different from the usual channel between the RI and devices. The security properties of OMA's messages are therefore unchanged in this mode.

4 Option 2: Heimdall as a Domain Member

In this method, we will allow Heimdall to have access to the domain key; that is, Heimdall is itself a member of the domain. Heimdall is then able to provide the domain key to a user device by encrypting the domain key with the public key of this device. An example sequence of messages using this option is shown in Figure 9.

The main advantage of this method is that during the joining of the domain, RIs need not be responsible for encrypting the domain key with the public key of the user device. However, we now require that Heimdall meet OMA's tamper-resistance standard since it has the domain key and has the ability to decrypt ROs obtained from RIs.

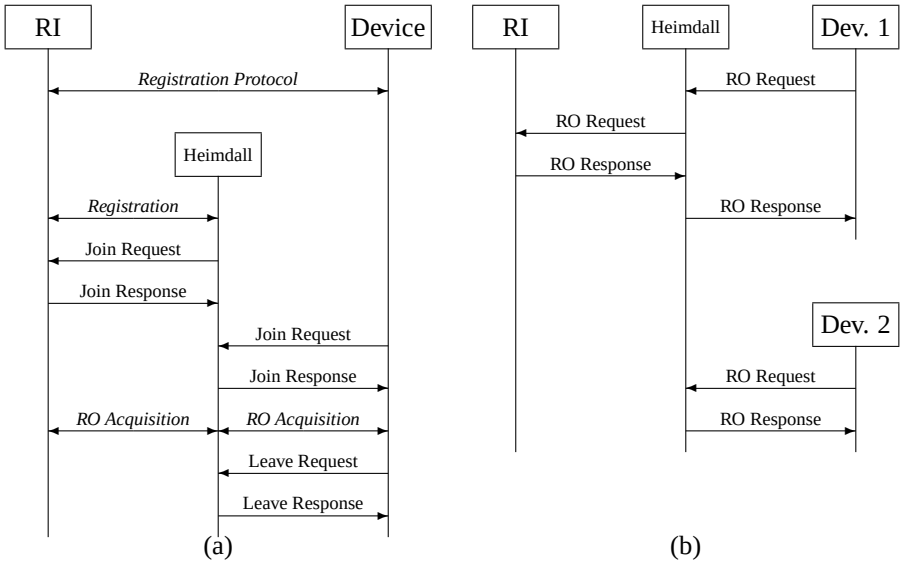

Figure 9: (a) Creating, joining and leaving domains for Option 2 and (b) acquiring a rights object.

4.0.1 Initialisation.

Devices are registered directly with RIs using the Registration Protocol. Heimdall is then registered using the same protocol, and the list of devices that have enrolled with the RI for this domain is passed to Heimdall in the *Extensions* field of the Registration Response. Heimdall then executes the Join Domain Protocol with the RI in order to obtain the domain key.

4.0.2 Joining Domains.

Devices join a domain by executing the Join Domain Protocol with Heimdall as shown in Figure 10. Heimdall verifies the identity of the device using the registration details provided by the RI, and constructs a positive Join Domain Response by encrypting the domain key with public key of the requesting device. The original Join Domain Response obtained by Heimdall from the RI is appended to the *Extensions* field of the new message, and the message is signed by Heimdall before returning the response to the device.

4.0.3 RO Acquisition.

Devices may acquire ROs by executing the RO Acquisition Protocol with Heimdall as shown in Fig. 11. If Heimdall does not have a copy of the requested RO, it obtains it from the RI using the normal RO Acquisition Protocol and caches it as in Option 1.

Heimdall → RI:	`Heimdall ID, RI ID, Heimdall Nonce, Request Time, Domain ID, Heimdall Cert. Chain, Extensions, Heimdall Signature`
RI → Heimdall:	`Status, Heimdall ID, RI ID, Heimdall Nonce, Domain Info. RI Cert. Chain, RI OCSP Resp., Extensions, RI Signature`
Device → Heimdall:	`Device ID, RI ID, Device Nonce, Request Time, Domain ID, Device Cert. Chain, Extensions, Device Signature`
Heimdall → Device:	`Status, Device ID, RI ID, Device Nonce, Re-encrypted Domain Info., Heimdall Cert. Chain, Heimdall OCSP Resp., RI Response, Heimdall Signature`

Figure 10: Option 2 Join Domain Protocol, including the initial joining of Heimdall to the domain.

Heimdall replaces the device nonce in the RO Response from the RI with the device nonce supplied by the requesting device, and appends the original RO Response to the extensions field of the re-written message. Heimdall then replaces the the RI's certificate chain and signature with its own and forwards the re-written response to the device.

4.0.4 Leaving Domains.

The devices leave the domain by executing the OMA Leave Domain Protocol with Heimdall.

4.1 Security

Device registration is identical to that used in the base OMA specification. The initial execution of the Join Domain Protocol between Heimdall and the RI is identical to the standard procedure for joining a device to a domain.

Given that Heimdall is trusted to check requests to join a domain against the list of devices supplied by the original RI, it is not possible for a device to join a domain unless it has been approved by the RI. This is the same as the base OMA specification.

Theoretically, devices can establish trust in Heimdall by following the certificate chain provided in the re-written Join Domain and RO Responses. Given that the device trusts Heimdall, it can be assured that the response was approved by a genuine RI. Devices can also check the original response by examining the extensions field of the re-written response.

Device → Heimdall:	`Device ID, Domain ID, RI ID, Device Nonce, Request Time, RO Info, Device Cert. Chain, Extensions, Device Signature`
Heimdall → RI:	`Heimdall ID, Domain ID, RI ID, Heimdall Nonce, Request Time, RO Info, Heimdall Cert. Chain, Extensions, Heimdall Signature`
RI → Heimdall:	`Status, Heimdall ID, RI ID, Heimdall Nonce, Protected ROs, RI Cert. Chain, RI OCSP Resp., Extensions, RI Signature`
Heimdall → Device:	`Status, Device ID, RI ID, Device None, Protected ROs, Heimdall Cert. Chain, Heimdall OCSP Resp., RI Response, Heimdall Signature`

Figure 11: Option 2 RO Acquisition Protocol

Standard OMA devices, however, may expect to find the certificate chain of the original RI in the responses and be confused by finding Heimdall's instead. The exact behaviour of a device may vary from implementation to implementation since the OMA specification does not specifically consider the case in which a valid certificate chain is provided, but is for a different entity than the one that originally issued the RO.

5 Option 3: Heimdall as a Rights Issuer

In this method we extend the responsibilities of Heimdall to registration of devices on behalf of the RI. This reduces the amount of traffic between Heimdall and the RI. An example sequence of messages using this option is shown in Figure 12. In this method, the RI and the user devices cannot authenticate each other directly and thus have to completely trust Heimdall.

5.0.1 Initialisation.

Devices are registered with Heimdall using the Registration Protocol. Heimdall registers with any RI it wishes to use using the Registration Protocol, and at the same time it provides the details of all of the devices registered to it so that the RI can charge the users accordingly. Heimdall then joins a domain and obtains the domain key from the RI that created the domain using the Join Domain Protocol as normal.

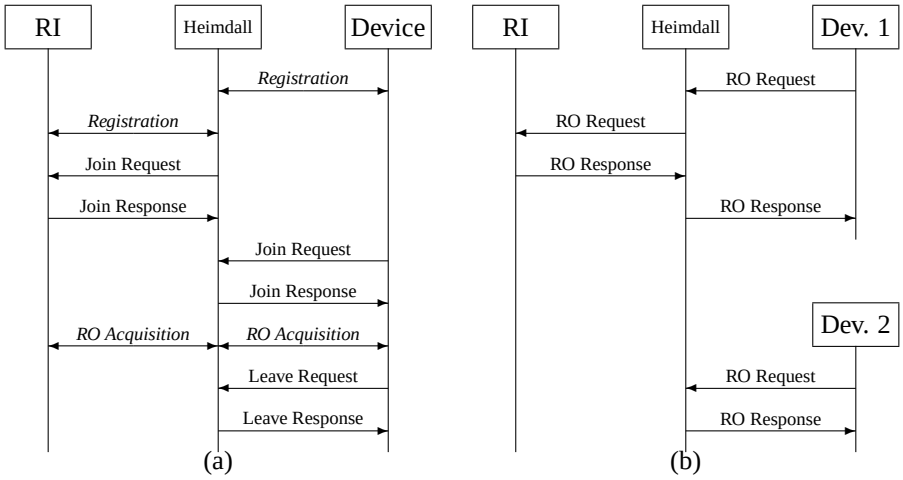

Figure 12: (a) Creating, joining and leaving domains for Option 3 and (b) acquiring a rights object.

5.0.2 Joining Domains.

Devices join the domain by executing the Join Domain Protocol with Heimdall. As in Option 2, Heimdall re-encrypts the domain key using the public key of the incoming device, and transmits this to the device using the Join Domain Response message as if Heimdall were a normal RI.

5.0.3 RO Acquisition.

User devices request and obtain content and ROs from Heimdall using a similar process to that used in Option 2. Heimdall obtains ROs from the original RI by executing the RO Acquisition Protocol as if it were a normal OMA device. Heimdall then replaces the RI's signature on the RO with its own signature, which is recognised by the devices that have registered with Heimdall. Devices that have registered to Heimdall can then obtain the RO by executing the RO Acquisition Protocol with Heimdall as if it were a normal OMA RI.

5.0.4 Leaving Domains.

Devices leave the domain by executing the Leave Domain Protocol with Heimdall.

5.1 Security

The relationship between Heimdall and user devices in this mode is identical to that between the RI and user devices in the base OMA specification. If Heimdall behaves identically to the original RI given the same input, it is easy to see that the system has

	OMA DRM	Option 1	Option 2	Option 3
Heimdall tamper-resistance	No	No	Yes	Yes
Heimdall domain policy	No	No	No	Yes
Heimdall load	None	Storage	Re-write	Total
RI load	Total	RO issue	RO issue	RO issue
Latency	None	Low	High	High
OMA Compliant	Yes	No	Unclear	Yes

Table 1: Comparison between OMA DRM and the three options discussed here.

identical security properties to those of the base OMA specification. To ensure this, it is necessary to assume that Heimdall has access to the domain membership policy supported by the RI, cf. [3].

If all domains have the same policy for admitting members (e.g. "at most n devices may be in the domain at any one time"), this policy can be coded into Heimdall at the time it is manufactured. If RIs support more than one kind of domain policy, the RI must communicate its policy to Heimdall as part of the *Extensions* field in the initial Join Domain Response sent to Heimdall.

Given that Heimdall is trusted to follow the domain policy, it is easy to see that it will behave exactly like the original RI and the domain will operate exactly as if it were managed directly by the RI.

6 Discussion

Table 1 gives a summary of the features of each of the three options we have discussed, and compares these (where applicable) to the original OMA DRM system. We summarise

- whether or not Heimdall is required to be tamper-resistant
- whether or not Heimdall is required to implement a domain policy
- the computational load placed on Heimdall;
- the computational load placed on the RI;
- the increase in latency caused by inserting the intermediary; and
- whether or not the mode can support standard OMA devices.

6.1 Other Domain Management Frameworks

The notion of a domain manager is also used in authorised domain frameworks proposed by Koster, et al. [3], Popescu, et al. [10] and Marlin [5]. The systems proposed by Koster, et al. and Popescu, et al. provide broadly similar functionality to

that provided by the Heimdall framework, but are not implemented within a standardised framework such as OMA. The Marlin specification was not available for public discussion at the time of writing.

The authorised domain frameworks proposed by Thomson [1] and the TIRAMISU Project [6] distribute the domain key to domain members by use of smartcards. The xCP framework proposed by IBM [9] is similar in that no nominated domain manager is required, but domain members distribute the domain key amongst themselves using a peer-to-peer protocol. These frameworks, however, are designed to support only household-type domains and it is not clear how well they would scale to larger domains.

7 Conclusion

We have described *Heimdall*, a domain management system for interacting with multiple RIs. The proposed framework provides the ability for users to join the domain, obtain content, transfer content between domain members and leave the domain. The introduction of Heimdall reduces the amount of computation performed by RIs, and reduces and the traffic between user devices and RIs.

We have compared the trade-offs made in three different modes in which Heimdall could operate, and shown that each mode can implemented so as to have the same security properties as the base OMA specification.

8 Acknowledgements

This work was partially funded by the Smart Internet Technology Co-operative Research Centre, Australia. We would particularly like to thank members of the Content Management Group at Telstra Research Laboratories for stimulating discussion in this area.

References

[1] J.-P. Andreaux, A. Durand, T. Furon, and E. Diehl. Copy protection system for digital home networks. *IEEE Signal Processing Magazine*, 21(2):100–108, 2004.

[2] Content Management License Administrator. Content Management License Administrator. `http://www.cm-la.com`, 2006.

[3] P. Koster, F. Kamperman, P. Lenoir, and K. Vrielink. Identity based DRM: Personal entertainment domain. In *IFIP Conference on Communications and Multimedia Security*, pages 42–54, 2005.

[4] D. W. Kravitz and T. S. Messerges. Achieving media portability through local content translation and end-to-end rights management. In *ACM Workshop on Digital Rights Management*, pages 27–36, 2005.

[5] Marlin Developer Community. Marlin – core system specification version 1.2. http://www.marlin-community.com, 12 April 2006.

[6] B. Marušič, P. de Cuetos, L. Piron, and Z. Lifshitz. TIRAMISU: That's unobtrusive DRM in the home domain. *Indicare Monitor*, 2(5), July 2005. http://www.indicare.org/tiki-read_article.php?articleId=125.

[7] M. Myers, R. Ankney, M. Malpani, S. Galperin, and C. Adams. X.509 Public Key Infrastructure Online Certificate Status Protocol – OCSP. RFC 2560, 1999.

[8] Open Mobile Alliance. OMA DRM v2.0 approved enabler, 3 March 2006.

[9] F. Pestoni, J. B. Lotspiech, and S. Nusser. xCP: Peer-to-peer content protection. *IEEE Signal Processing Magazine*, 21(2):71–81, 2004.

[10] B. C. Popescu, B. Crispo, A. S. Tanenbaum, and F. L. A. J. Kamperman. A DRM security architecture for home networks. In *ACM Workshop on Digital Rights Management*, pages 1–10, 2004.