17

An Introduction to Digital Rights Management Systems

Willem Jonker

Philips Research / Twente University
The Netherlands

Summary. This chapter gives a concise introduction to digital rights management (DRM) systems by first presenting the basic ingredients of the architecture of DRM systems for (audio and/or video) content delivery, followed by an introduction to two open-standard DRM systems, one developed in the mobile world (Open Mobile Alliance DRM) and another one in the world of consumer electronics (Marlin).

17.1 Introduction

Digitization and new coding formats, together with the development of the Web, have led to a world where electronic distribution of audio and video to end users has become a reality. This reality however has at the same time led to increased concern about the protection of the rights of owners of the content that is distributed in electronic form. Digital right management (DRM) is the term for commercial, legal, and technical measures that can be taken to ensure that rights of owners of digital content are respected. So, DRM is more than technology, DRM can only function in a legal framework that includes legislation, conformance, enforcement, etc. In this chapter we will concentrate on the technical aspects of DRM systems. For a broader overview we refer to [1,2,3].

From the technical perspective there are two main approaches towards the application of DRM in the context of audio/video delivery; one is based on watermarking and the other is based on cryptography. Both approaches will be discussed in the following chapters. In this chapter we concentrate on systems that are based on cryptography.

Currently some proprietary DRM systems are out in the market, such as Apple FairPlay and Microsoft DRM, while other open-standard DRM systems, such as Open Mobile Alliance (OMA) DRM v2 and Marlin are under development.

17.2 DRM Systems: The Basics

The figure below shows the typical architecture of a DRM system for electronic delivery of audio. On the left are the components that normally reside on the server side, while on the right are the components that reside on the client side.

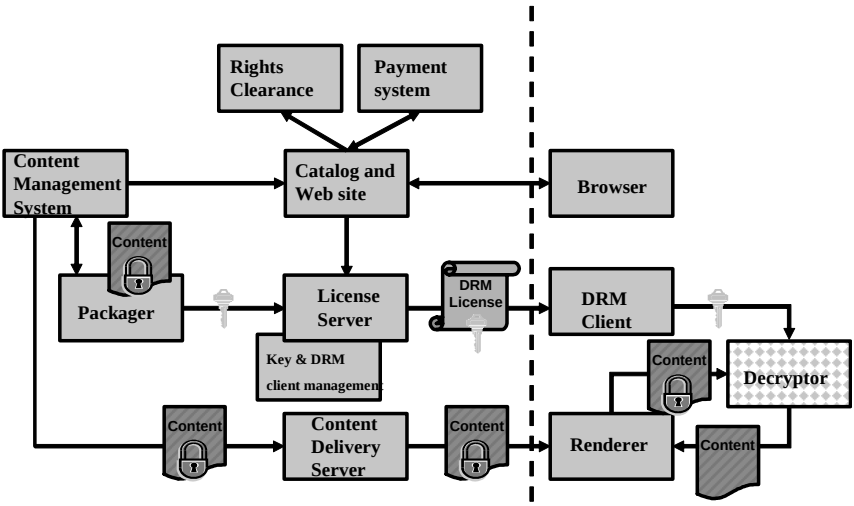

Fig. 17.1 A generic DRM Architecture for audio delivery

A typical usage scenario is one where the user uses the browser to select a specific audio item to be acquired. The browser connects to a catalogue Web site where the items are listed. When a specific item is purchased, a payment has to be handled, which is done by a payment system. At the same time the rights that the user wants to obtain need to be issued. Once this is done, the content management system is instructed to prepare the audio item for shipping. This includes encrypting the audio. This task is carried out by the packager, which sends the encryption key to the license server. The latter is responsible for creating the appropriate license[1]. Often the encryption key is part of the license. Once the content is properly packaged it is shipped to the delivery manager, which sends it to the client. At the same time the license manager sends the license to the client.

[1] Although often used intermixed, there is a difference between rights and licenses. Licenses are the actual carriers of the rights (in the form of rights expressions) and often contain additional information, most notably the content encryption keys.

At the client side the DRM client uses the license to decrypt the audio and controls usage. After decryption the audio is send to the renderer and played.

The following elements are the core ingredients of a DRM system:

1. The content to be protected. The content that needs to be protected can vary. DRM is currently most known from audio and video content delivery, but DRM is also applied to documents (e.g. Adobe) and can in principle be applied to any form of digital content.
2. The cryptographic protection scheme, including a key management scheme. Most DRM systems work with a combination of symmetric and asymmetric keys. Often content is encrypted in layers using a key hierarchy, where keys in the lower layers in the hierarchy are used to encrypt parts of the content.
3. The expression of the rights. In most systems rights expression languages are used to define the rights that are issued to the content users. Two well-known rights languages are ODRL[4] and XrML [5]. DRM rights languages are often expressed in XML. Since for digital assets in general there are a lot of possible situations, DRM rights languages tend to be complex.
4. License management. The license contains both the encryption key as well as the rights that have been entitled to the user. Without the presence of the license, content access is not possible. Since the license contains both the key and the rights it is important to protect the license as well. It should be impossible for an unauthorized user to get the key from the license or to change the rights expressions. Licenses should therefore be handled with care and either be encrypted or stored in a secure place in the client. Some systems support the explicit separation of content and licenses. This has the advantage that licenses and content can be send over different channels and at different times.
5. Compliance regulation. Compliance is a key issue in DRM systems. Only compliant devices can participate in content exchange. A compliant device is a device that respects the rules of the DRM system. This means that the device guarantees that the content is treated as described by the license rights and that the device also takes certain measures to prevent encryption keys from being obtained by unauthorized users. In the (open) PC environment compliance is often only supported in software by installing DRM client software. In the (closed) CE environment compliance is often supported by a combination of software and hardware. A way to deal with compliance is using certificates. For example a CE device may have a pre-installed certificate indicating that it is compliant with a specific DRM system. Upon request, this certificate can be send to a license server in order to verity compliance. A certificate authority that monitors compliance and acts in the case of violations issues such certificates. Certificates that have been issued to compliant de-

vices that nevertheless violate the rules can be revoked. Revoked systems will not be able to acquire content under the DRM scheme any longer.

6. Client-side enforcement. The enforcement of rights at the client side is the most challenging part of a DRM system. In principle the task is simple: at the client side specific DRM client software is installed that intercepts the access to the content and enforces the rights. However the challenging part is the prevention of attacks. The client side is not under the control of the DRM system and is therefore vulnerable to attacks. These attacks may be aimed at disclosing the encryption key or at circumventing the enforcements of the rights. In order to avoid such attacks the DRM client needs to be executed in a tamper-resistant environment with protected memory and isolated execution of the DRM client code. In the PC domain the trusted computing group [6] is working on tamper-resistant environments; this work has led for example to the specification of a TPM [7].

This concludes our discussion on the general basic aspects of DRM systems. In the remainder of the chapter we will focus on a number of concrete open-standard DRM systems.

17.3 OMA DRM

The Open Mobile Alliance (OMA) [8] is working on architectures to enable all kinds of value-added services over mobile networks. One class of services that is targeted is that of content download services. These services require content protection and for that reason OMA has developed a DRM specification, called OMA DRM. Actually there are two specifications out today: OMA version1 and OMA version2. We will briefly describe OMA version 1 and than concentrate on OMA version 2.

17.3.1 OMA DRM Version 1

OMA DRM v1 can hardly be called a DRM system, since it lacks almost all protection one would expect from a DRM system. The security of OMA DRM v1 depends completely on the security of the underlying mobile network. Over this network OMA v1 messages are sent in the clear and as a result OMA DRM v1 is very vulnerable to attacks.

OMA DRM v1 has three different content delivery schemes:

1. Forward lock
2. Combined delivery
3. Separate delivery

In the forward lock scheme, the content is downloaded by means of the download mechanism offered by the WAP protocol. The content is not encrypted but sent in the clear. Together with the content a wrapper is sent that indicates that the content may not be forwarded to other mobile devices. The OMA DRM v1 client implementation checks the wrapper and refuses to forward the content. As can easily be seen the system is vulnerable to several attacks. The content is not encrypted and thus can be read from the mobile device memory. Also the wrapper is not protected and thus can be easily changed. Finally, the DRM client is not protected and thus the intended operation can be easily circumvented.

The combined delivery scheme is to a large extent similar to the forward lock scheme. However, instead of sending the content with a wrapper, now the content is sent with a license that contains a simple rights description indicating what the user is allowed to do with the content. This scheme is vulnerable to the same attacks as the forward lock scheme.

Finally, in the separate delivery scheme the content and the license are downloaded to the client in separate messages. In this scheme the content is encrypted and downloaded using the download mechanism of the WAP protocol. The license now contains both the rights expression and the content key, which is used to encrypt the content. The license is sent to the client using an unconfirmed push mechanism from the WAP protocol. Although some security is added, the separate delivery scheme is still very vulnerable to attacks. The license can easily be intercepted due to the fact that the WAP push is not confirmed. Also the DRM client is still not protected.

17.3.2 OMA DRM Version 2

For more-serious content delivery services OMA DRM v2 has been developed. OMA DRM v2 follows very much the basic DRM architecture as described above. It distinguishes between a content issuer and a rights issuer as two separate entities. Rights are only issued to trusted DRM agents that reside in client devices (e.g., mobile phones).

Compliance in OMA DRM v2 is realized through the notion of trusted agents. The notion of a trusted agent is important and implemented by means of a public key infrastructure. This means that each trusted agent has a unique public/private key pair and a certificate belonging to it. Certificates play a role in the authentication protocol and are a means to revoke agents that exhibit noncompliant behaviour. Content is encrypted by means of symmetric keys, while licenses (or rights objects as they are called in OMA) are encrypted by asymmetric keys. The symmetric key belonging to some specific encrypted content is sent to a trusted DRM agent by means of a rights object. The rights object is encrypted with the public key of that specific trusted DRM agent. In this way only that specific trusted DRM agent can access the rights object by decrypting it with its

corresponding private key. This is a way to bind content to a specific device and to prevent that content from being played on another device[2].

In order to allow some sharing of content OMA DRM v2 introduces the notion of domains. A domain is a group of devices defined by a rights issuer. Domains are optional and their use can differ among various content issuers and rights issuers. A rights issuer carries out domain management. In order to join a domain, a device has to register to that domain by making a request to the rights issuer. Once a group of devices has joined the domain, they can access the content that is issued to that domain. This means that devices can directly share rights objects for domain content amongst each other.

In addition to domains, OMA DRM v2 supports another mechanism for sharing content, namely super-distribution. Super-distribution can be used between any two trusted OMA DRM agents. It consists of sending the protected content from one agent to the other. For the other agent to gain access to the content, it has to contact the rights issuer in order to obtain a rights object for that specific content. The nice thing about super-distribution is that it allows direct exchange of the protected content and allows for the rights object to be acquired later. In a mobile environment this may be an advantage since rights objects will be much smaller than content objects. So a user may acquire the content through a fast direct connection with another mobile device, while obtaining the rights object via the mobile network. This is a direct result of the decision in OMA DRM v2 to separate content objects and rights objects.

The OMA DRM v2 rights expression language (REL) is a subset (or so-called mobile profile) of ODRL v1.1. The REL is an XML structure. It is beyond the scope of this chapter to give a detailed description of OMA DRM v2 REL.

As far as client-side enforcement is concerned, OMA requires the secure storage of the private device keys and the implementation of a secure clock in connected devices. In addition it is required that the execution of rights evaluation at playtime is secured and cannot be tampered with. The reason for requiring a secure clock is to support time-based usage rights (for example the right to use the content up to a certain date) and to prevent users from manipulating the clock in order to affect the impact of time-based rights.

[2] Binding content to devices is typical for so-called device-based DRM systems. Device-based DRM systems are limited in sharing content over various devices even when these are owned by the same end-user. In the section on Marlin we will see how so-called person-based DRM systems try to avoid this limitation.

17.4 Marlin

While OMA DRM originates from the mobile world, Marlin originates from the CE world. The core developers of Marlin [9] are InterTrust, Sony, MEI, Samsung, and Philips. Marlin is an open DRM standard targeting CE devices and supporting the controlled flow of audio and video content over collections of CE devices.

Marlin has a number of characteristics that differentiate it from other DRM systems. We will list them first and then elaborate on them below. Most important is that Marlin is user-based, rather than device-based, which means that licenses are bound to users rather than to devices. A second characteristic that differentiates Marlin is that it does not use a rights expression language, instead rights definition and enforcement in Marlin are taken care of by means of a control program. Such control programs are part of the generic DRM architecture called Octopus. A third characteristic of Marlin is that right from the start the notion of domain is designed in. The Marlin domain model builds on a graph of nodes and links that allow for very flexible rights sharing.

The overall Marlin architecture consists of four classes of actors: the Marlin client, the Marlin domain manager, the Marlin registration service, and the Marlin license service. The Marlin client has the same role as other DRM clients: control the access to the content based on the rights that have been issued to the user. The Marlin domain manager has the role of managing domains consisting of devices and users joining and leaving domains. The Marlin registration service is responsible for admitting users and devices to the Marlin system; it does so by means of issuing nodes and links. Finally, the Marlin license service issues licenses.

Nodes and links play a central role in Marlin. Nodes represent entities in a Marlin system. There are four kinds of nodes: device, domain, user, and subscription. In Marlin, links express an inclusion relationship.

The directed graphs play a central role in determining the access rights to content in a Marlin system. Roughly speaking content can be accessed when there is a path in the graph from the requester to the content. Note that this a pre-requisite, the actual access rights are expressed in the control program, the graph serves as a sharing mechanism that allows sharing of licenses between users and devices in a very flexible way.

17.4.1 Marlin: User-Centric

As stated before Marlin is user-based, which means that licenses are bound to users rather than to devices. Binding a license to a user in an actual Marlin implementation means that the content to which the license refers is encrypted with the *public* key belonging to that user. The *private* key of the user has to be stored somewhere in a secure place and processing of

that key needs to be done in some secure environment. This can be on a token, on a device linked to the user, or somewhere in the network. When a user wants to play its content on a device, in some way a connection needs to be established between the device and the private key of the user, allowing for the decryption of the content on that device. As well as user-binding, graphs play a role in the sense that the license can contain additional requirements on the graph, for example that there is a path from the device to the user. The enforcement of such requirements is done by means of the control programme that is part of the license.

17.4.2 Marlin Domains

The domain concept in Marlin is very flexible and offers all kinds of ways to share content in a controlled way. Both users and devices can join a domain. By releasing content to a domain the content is made accessible to all devices that are part of that domain, and as such the content can be freely shared among the devices and (when authorized) users. Domains are very dynamic in various ways. New domains can be created, domains can be removed, devices can join and leave domains, and users can be (de)associated. In order to prevent the world as a whole from becoming one big domain in which everybody can freely exchange content, there are restrictions on the number of devices that can be members of a domain as well as restrictions on the number of domains a user can join. By separating the domain structure from the licenses Marlin has become very flexible. Once a service provider has installed a domain policy, changes of domains do not affect licences any more, which allows domain modifications (within the policy) without affecting licenses.

17.5 Concluding Remarks

In this chapter we have presented two open-standard approaches towards digital Rights Management. In addition to the open-standard approaches there are quite a number of commercial DRM solutions on the market of which Microsoft DRM and FairPlay from Apple are the best known. Especially this first generation of commercial DRM systems that is currently deployed still has a long way to go in terms of convenience to the end user. Important issues that need to be addressed are DRM interoperability, ubiquitous access to one's own content at any place at any time, sharing and gifting of content, as well as hiding the DRM complexity from the end-user. In the following chapters some of these issues will be addressed in more detail.

References

1. B. Rosenblatt, B. Trippe, S. Mooney, *Digital Rights Management*, Business and Technology, M&T, New York, 2002.
2. E. Becker, W. Buhse, D. Gunnewig, N. Rump, "Digital Rights Management: Technological, Economic, Legal and Political Aspects", LNCS 2770, Springer, Berlin-Heidelberg, 2003.
3. W. Jonker, J.P. Linnartz, "Digital Rights Management in Consumer Electronics Products", IEEE Signal Processing Magazine, Vol. 21, Issue 2, pp. 82-91, March 2004.
4. R. Iannella, Open Digital Rights Language (ODRL) Version 1.0, IPR System Ptd Ltd. November 2001. Available at: http://odrl.net/1.0/ODRL-10-HTML/ODRL-10.html
5. XrML - The Technology Standard for Trusted Systems in the eContentMarketplace. Available at: http://www.xrml.org/
6. www.trustedcomputinggroup.org
7. www.trustedcomputinggroup.org/specs/TPM
8. www.openmobilealliance.org
9. www.marlin-community.com