Core Security Requirements of DRM Systems

H.L. Jonker[1,2] and S. Mauw[2,1]

[1] Eindhoven University of Technology
Department of Mathematics and Computer Science
P.O. Box 513, NL-5600 MB Eindhoven, the Netherlands

[2] University of Luxembourg
Computer Science and Communications Research Unit
6, rue Richard Coudenhove-Kalergi, L-1359 Luxembourg, Luxembourg
h.l.jonker@tue.nl, sjouke.mauw@uni.lu

Abstract. The use of Digital Rights Management (DRM) systems involves several stakeholders, such as the content provider, the license provider and the user, each having their own incentives to use the system. Proper use of the system implies that these incentives can only be met if certain security requirements are fulfilled. Quite some attention in literature has been devoted to specific security aspects of DRM systems. The contributions of this chapter consist of a systematic overview of core security requirements for DRM systems. This chapter conducts a stakeholder analysis and develops a simple, generic conceptual model to arrive at such a set of core security requirements.

Keywords: Digital rights management, security requirements.

1 Introduction

There is a precarious balance between dissemination of information (to the general public) and stimulation of innovation and art. The easier it is to spread new information, the less possibilities to profit there will be for innovators to reap the fruits of their labor. On the other hand, spreading innovation and art is considered beneficial to society.

The introduction of computers has had a profound impact on this balance. With computers, it is trivial to create a perfect copy of *content* – a term used to indicate a work of art, such as msuic, literature, movies, etc. This coupled with the widespread availability of broadband internet connections means that completely new venues for spreading content to the public at large have come into existence. This enables a business model, that consists of selling and delivering digital versions of content online. The main point of concern for such a business is to prevent unsanctioned redistribution of the delivered content.

Digital Rights Management (DRM) systems have been created for this goal. The purpose of a DRM system is to protect (digital versions of) content. Content is bound to a license, and the content is only accessible under the terms stated by the license.

In recent years, there has been a strong push into the research and development of DRM systems. There has been work on various related security aspects such as secure storage [21], traitor-tracing [12, 18], watermarking [4], fingerprinting [6, 16] and tamper resistant code [8, 1]. There have also been various proposals for models of DRM systems with specific properties [5, 14, 13, 20].

These proposals incorporate various security requirements. Some of these requirements assure core DRM functionality, whereas other requirements realise the specific properties for which that architecture was constructed (e.g. interoperability: MOSES [20], Coral [3]). The emphasis of such proposals is usually on the latter type of requirements. It is not uncommon that the requirements assuring core DRM functionality receive a lesser treatment. These requirements are often not all made explicit, nor is a justification for them provided. Which of these requirements are made explicit varies from proposal to proposal, which means that the set of requirements that assure core DRM functionality is scattered.

There are several reasons to make this core explicit. The first and foremost reason is that security is an enabling factor for DRM systems. DRM systems are designed to provide a solution for a security problem. An understanding of (the justification for) the core security requirements is crucial for fundamental comprehension of the security of DRM systems.

Moreover, knowledge of the core security requirements is instrumental in the construction and verification of DRM systems. Such knowledge enables developers to better understand the consequences of security trade offs. In practical systems, such trade offs between desired features and security requirements are not uncommon.

For example, Apple's iTunes allows the user to create a CD of protected music. Naturally, Apple realised that such a CD could be used to copy music. Nevertheless, this feature was deemed more important than the costs in terms of loss of security. In this case, an informed decision has been made. In other respects, some of the design decisions of iTunes seem less well-informed and have a negative impact on the overall security of the system. A more detailed examination of iTunes follows in Section 4.3.

This chapter uses a structured approach to identify core security requirements and provide a justification for them. The goal of this chapter is focused on the security aspects of the design of DRM systems and our aim is to systematically derive core security requirements for DRM systems. Although there is a wealth of methodologies supporting system analysis and design, the methodologies for deriving security requirements are only at their infancy. Therefore, our research will start by identifying some useful methodologies and combining their strengths.

In order to provide a base for the found security requirements, we describe a model which limits itself to the core processes of a DRM system. The generic nature of this core model of DRM functionality implies that requirements found for it are applicable to most DRM systems. The extensibility of this core model indicates that it can be augmented to accommodate additional functionality, and therefore that this model suffices for our needs.

The rest of this document is organised as follows: Section 2 details the approach we used to arrive at our security requirements, resulting in a list of security properties and a generic process model of DRM systems. These form the basis of the security requirements in Section 3. Section 4 examines the practical application of this research. And finally, we present some conclusions in Section 5.

Acknowledgments. The comments and commentary of ir. Lex Schoonen and ir. Jan Verschuren were invaluable during our research, and we are grateful for their constructive commentary. We would also like to thank Paco van der Linden and Jilles Tjoelker for sharing their investigations of security risks of Apple iTunes and Microsoft Windows Media DRM, respectively.

2 Problem analysis

To establish the core security requirements of DRM systems, we performed a problem analysis. The analysis led to a terminology and a description of the desires of the various involved parties. These desires are used in Section 3 as the foundations for the security requirements of the system.

The problem analysis consisted of three steps. The first step established core stakeholders, their incentives and relevant terminology of DRM systems. The second step consisted of deriving the desired security properties from the incentives and the terminology. The third step consisted of deriving a process model. The process model is to capture the core operations occurring in DRM systems.

2.1 Establishing terminology, stakeholders and their incentives

The first step in deriving the security requirements is a *stakeholder analysis*. The purpose of this step is to determine the individuals (or roles) who have an interest in using a DRM system, and

their incentives for participating. This understanding of their incentives is important, as these incentives lead to security requirements.

To establish the stakeholders and their incentives, a method similar to and inspired by several existing methodologies from the field of Information Systems has been used. We based our research on a variety of methodologies, such as domain analysis (see e.g. [17]), stakeholder analysis (for an overview see [15]), system decomposition (see e.g. [22]). Normally, these methods assist in designing a system. However, our goal was not to design a system, but to focus upon a system's security aspects. Parts of these methods were accordingly adapted to capture the security aspects of DRM systems.

The problem analysis leads to the following list of stakeholders and their roles in a DRM system:

stakeholder	role
media company	creates content
developer	creates DRM systems
user	acquires content
network provider	transports content
reseller	acts as an intermediary between users and media companies

Each of these stakeholders has their own, specific incentives for using DRM. DRM systems deal with providing users access to content. This observation together with a generalisation of the stakeholders' roles leads to the following formulation of the core functionality of a DRM system:

> *A DRM system is a system that allows* users *to access* digital content *according to access conditions as specified in a* license*. Licenses and content are provided by* license creators*, who in turn acquire content and generic access conditions from* content creators*.*

The above description distinguishes three types of core participating roles: the content creator, who creates content; the license creator (or reseller), who creates specific licenses; and the user[3], who desires access to content. Although the other roles are essential for an operational DRM systems, we will not study their security requirements in this chapter. Network security and e-commerce security are out of the scope of the current description.

The analysis indicates that the roles of content creator and license creator are executed by companies (e.g. media companies). As companies are legal entities, they are firmly embedded in a legal framework and thus there exist numerous non-technical solutions to ensure that companies adhere to access agreements. This observation holds less for individual persons, which allows for an asymmetry in DRM systems: there are less deterrents to prevent individuals from circumventing agreements then there are deterrents preventing companies. Hence for individuals strict technological enforcement of access agreements between them and companies is necessary. This leads to a natural tendency in DRM development to focus on enforcing access control at the user's side, and to focus less (or not at all) on access control at the license creator's side.

Each of the core roles has various reasons for taking part in a DRM system, and thus related security concerns. The following outlines their incentives:

Content creator The role of content creator is executed by stakeholders that create content, such as media companies. They do not seek interaction with users directly, but are satisfied with leaving this to an intermediary – i.e. license creators. DRM can be used to enforce the conditions set on content by the content creator for the license creator. However, this can also be dealt with by non-digital means (contracts, agreements between companies).
Content creators use DRM technology to support new business models. For instance, they can create a bundle of desired content and other content. Such a bundle could increase the value of the content for users (e.g. by including the 'making of' footage), or it could increase revenue for the content creator (e.g. by including commercials). Additionally, using DRM technology it

[3] We prefer the computer science term 'user' over terms originating from other fields, such as 'client', 'customer' and 'consumer'

is possible to offer a revenue-generating alternative for traditional downloading. This means that DRM technology can open a new market. Finally, offering content online in a digital version means that the per-content overhead costs are low - there is no need for plastic casing, a colourful cover, etc.

License creator The license creator binds content to access conditions specified in a license. This role saves content creators the overhead of negotiating directly with their customers. The license creator can use DRM technology to offer tailor-made access to content to users. On top of that, overhead compared to selling physical media is substantially reduced, because digital content takes up little physical space and the presentation of the content can also be done digitally. By offering a clearly legitimate and known-quality alternative for downloading, license creator can open up a new market. And lastly, as content is bound to a license created by the license creator, access to content distributed in this way will comply with the access conditions set by the license creator.

User Users will be drawn to DRM systems because DRM systems offer a legitimate, known-quality alternative to more dubious sources of content. Ease of use is an important consideration in this regard: if use of the system or acquisition of content becomes bothersome, a user might turn back to other sources.
Another advantage that DRM systems can offer users is the possibility to restrict the access to (and thus the cost of) content to precisely what the user wishes.
For users, it is important that DRM offers an improvement upon aspects of existing content distribution channels, otherwise there is no incentive for users to switch to a DRM distribution channel. This can, for example, be in terms of ease of use or availability. Such improvements can be offset by deterioration of other aspects – privacy concerns could turn users away from a DRM system.

2.2 Security properties per core role

The second step of the problem analysis consists of deriving the desired *core security properties* of each core stakeholder, by combining the stakeholder analysis and the established terminology. The high-level properties of this section are then made precise as *concrete security requirements* in Section 3.

For each core role in DRM systems a translation of the stated incentives into desired properties is provided. There are various security solutions that come into play in this process (e.g. a payment infrastructure). We confine our examination to the core of a DRM system, i.e. those required by the core roles for basic functionality of a DRM system.

As the intent of this section is to establish which security properties are desired for DRM systems, properties outside the scope of DRM systems (such as those governing negotiations between parties, those ensuring that all parties uphold their end of contracts, those governing the privacy of the business operations of license creators, etc.) are not considered below. Properties of DRM systems that are not security related (e.g. functional properties) are also not examined. Lastly, we confine our examination to the core of a DRM system, i.e. the measures a DRM system provides for protecting digital content.

Content creator Of paramount importance for content creators is that the content they provide remains protected. Since digitising content facilitates widespread copying of content, content creators need to be assured that the DRM system that safeguards their content will only allow access to content to the right party, when conditions in the right license have been met.

Adherence of license creators to the terms set by content creators can also be dealt with outside the DRM system. As most current DRM systems focus on enforcing access control at the user's side, they delegate the specific security requirements of content creators to license creators. Hence, a requirement for adherence to the license suffices. More precisely:

c.1 Content is only accessible by untampered components created by the official system developers, under the conditions of a valid license issued by a bona fide license creator.

License creator The license creator acts as an intermediary between users and content creators. The content creator has delegated responsibilities for content protection to the license creator. The role of the license creator is to create a license, that specifies the conditions under which a user is allowed access. Naturally, a license creator expects the DRM system to enforce this license. Furthermore, it is in the license creator's interest to be able to deliver what users want, when they want it – otherwise they may opt for an alternative distributor. Finally, a license creator will also be impacted by attacks on the system, hence the effect of successful attacks on the system should be limited, and attacks disrupting or twisting the contact with users should be prevented.

The above is formulated more precisely in the list below:

l.1 Content is only accessible by a renderer with a valid license issued to that renderer, originating from the license creator, under the terms stated in that license.
l.2 The DRM system precisely delivers the content that has been requested, with the license as requested, in the correct format, at the desired time to the user.
l.3 The impact of breaking the system must be constrained.
l.4 Safe and secure negotiations with the user: other parties cannot influence (hack/break) the license creator's side of the negotiations

User Users have the option of using traditional distribution channels and using a DRM system. In order for a DRM system to appeal to users, it is important that a user has the impression that the benefits of using DRM outweigh any more negative aspects compared to traditional distribution (e.g. a purchase in a store). Users will thus expect to be able to download content anywhere, at any time. Like a purchase in a store, such a transaction can only occur with the active participation of the user. When buying content through traditional channels, users can remain anonymous (up to a certain degree) – in a digital setting it is much easier to link data to users, violating their privacy. Finally, in traditional distribution channels, a user is aware of the agreement (which content, accessible under which conditions, for how much money) she makes when purchasing content. This should also hold in a DRM setting.

Expressing these points exactly leads to the following list:

u.1 The user can precisely acquire a consumable form of the content that the user desires, at the moment the user desires it.
u.2 To order licenses or content on the user's behalf requires the intensional participation of the user.
u.3 Neither content nor licenses can be linked to the user.
u.4 The user is aware of all negotiations resulting in agreement between her and the license creator, and consents to the terms of any such agreement.

In Table 1 the desired security properties for each stakeholder are expressed using the established terminology. Together, these properties form a solid foundation for the core functionality of DRM systems – and therefore, they also provide sound underpinnings for the security properties of DRM systems. These properties are necessary for DRM systems. Sufficiency depends on the completeness of the descriptions in Section 2.1, and on correctness of the translation of these incentives into the expressed security properties.

Not all the security properties as mentioned in Table 1 will be applicable in all situations – specifically note the privacy property u.3. However, we believe that privacy is an important security aspect, especially in e-commerce systems. Therefore we feel that inclusion of privacy as a core incentive is justified.

content creator	
c.1	Content is only accessible by untampered components created by the official system developers, under the conditions of a valid license issued by a bona fide license creator.
license creator	
l.1	Content is only accessible by a renderer with a valid license issued to that renderer, originating from the license creator, under the terms stated in that license.
l.2	Precisely deliver what has been requested, in a consumable form, at the desired time for the licensee.
l.3	The impact of breaking the system must be constrained.
l.4	Safe and secure negotiations with the user: other parties cannot influence (hack/break) the license creator's side of the negotiations
user	
u.1	Precisely acquire a consumable form of the content that the user desires, at the moment the user desires it.
u.2	To order licenses or content on the user's behalf requires the intensional participation of the user.
u.3	Neither content nor licenses can be linked to the user.
u.4	The user is aware of all negotiations resulting in an agreement between her and the license creator, and consents to the terms of any such agreement.

Table 1. Security properties as desired per role

2.3 Conceptual process model

The second step of the problem analysis consists of the development of a *conceptual process model*. This model relates the basic processes in the DRM system to each other, and can be refined to provide a basis for identifying security requirements. This process model is then combined in Section 3 with the list of security properties to establish core security requirements of DRM systems.

To ensure that the model is applicable to as many DRM systems as possible, it must be as generic as possible. In order to derive such a generic model, we start with one component for the three core roles in a DRM system: the content creator, the license creator and the user, see Figure 1. Components for non-core roles are left out as they can be introduced when necessary, as such additions constitute refinements of the core model. The resulting model is subsequently refined to incorporate various specifics of DRM systems.

The model of Figure 1 is read as follows: the content creation component creates content, which is forwarded to the license creator. The license binding component binds a license to the content, after which content and license can be sent to the user. The content rendition component at the user side renders the content.

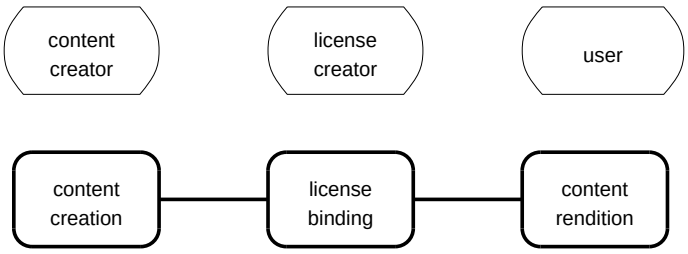

Fig. 1. Basic process model

The first refinement to this model is instigated by the observation that to be perused by users, the content is eventually transformed into an analogue form. This means that there exist two variants of the content: digital content (which the system is to protect), and analogue content (which a user can consume). As all DRM systems deal with both variants, the generic model can be refined to incorporate this distinction without harming its genericity. This results in splitting the content rendition component: one part to convert digital content to analogue content (the *content renderer*), and one part to communicate with the license creator's component (the *content processor*). This refinement is depicted in Figure 2. Note that in general, rendition is the final step in the process. Hence, the content renderer is depicted as the last process in the model.

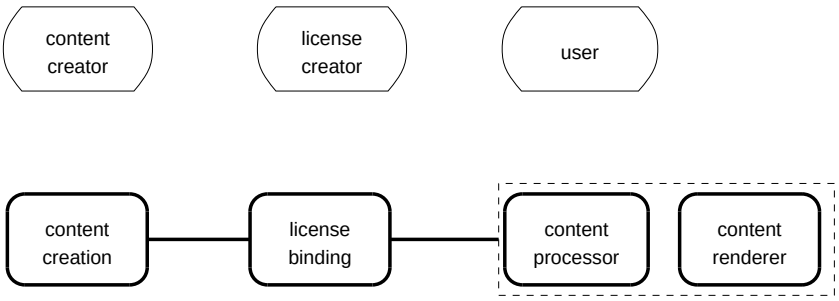

Fig. 2. Model incorporating rendering of content

The next refinement is prompted by noting that DRM-protected content cannot be accessed without a license. This does not imply that a license must be bundled with the content in all cases, or even that a license must exist when the content is protected. It suffices that the resulting content is not accessible without a valid license. To express the possibility of licenses and protected content existing separately, the license binding component is further refined into two parts. The *content protection* part provides the user with protected content, while the *license creation* part provides the user with a license for protected content. This leads us to the conceptual model depicted in Figure 3:

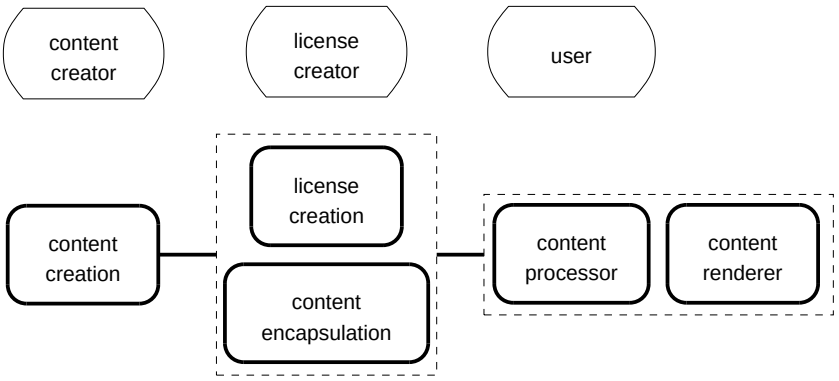

Fig. 3. Generic process model of DRM

This generic process model now incorporates the core roles and models the core processes in a DRM system. The model can be refined to comply with virtually all existing DRM architectures.

For example, additional roles like content issuer and license issuer can be incorporated to comply with the functional architecture of the Open Mobile Alliance DRM architecture [13].

The goal of this chapter is to establish core security requirements of DRM systems. As further refinements constrain the applicability of the model, the model is not further refined.

2.4 Results of the problem analysis

The problem analysis has provided two results: a list of security properties desired by the core stakeholders, and a conceptual process model of the processes taking place in a DRM system. Taken together, these results elucidate security requirements of the core functionality of a DRM system.

Completeness of these descriptions can have a large impact on which security requirements are found. After all, the more complete the given descriptions are, the more complete the derived security requirements will be. The methods we adapted to arrive at the given descriptions support a systematic derivation of security requirements, but do not guarantee completeness of the results. Despite this, we believe that, due to the systematic approach, the descriptions are sufficiently exhaustive for our goals.

3 Core security requirements

In this section, the incentives as described in Section 2.1 are used as a basis for establishing the security requirements per stakeholder upon the generic process model, as shown in Figure 3. In Section 2.2, they were exactified to describe the precise security properties desired by stakeholders. In this section, these resulting properties are translated into security requirements upon the model. Both the process model and the incentives were deliberately kept widely applicable, so that the resulting security requirements will also hold in general for DRM systems. To describe security requirements on a specific setting or DRM system, both the process model and the below requirements can be further refined. An example of such a refinement is the inclusion of the role of Certification Authority.

The refinement of security properties was based on a technique described by Schneier in [19]: attack trees. We adapted this technique to enable systematic analysis of defensive aspects of a system – in this case, the generic process model (Figure 3). The results of this systematic analysis are presented below.

3.1 Content creator

The desired security properties for the content creator were established as follows in the previous section:

c.1 Content is only accessible by untampered components created by the official system developers, under the conditions of a valid license issued by a bona fide license creator.

Note that there is a large overlap with the security properties desired by the license creator, particularly incentive l.1. As noted earlier, DRM systems can be asymmetric: the relationship between a content creator and a license creator is not on par with the relation between a license creator and a user. In many DRM systems, the content creator trusts the license creator to adequately see to the security of the system. To support this asymmetry in our generic security requirements, we have chosen to not mention security requirements twice (once for the content creator and once for the license creator), but to only formulate them for the license creator. As mentioned in the introduction, to model a symmetric DRM system it suffices to refine the requirements and the process model.

Ensuring this property holds for the generic process model, given that the license creator is trusted to comply with and ensure the security requirements, limits the security requirements to requirements upon the communication with license creators. If only trusted license creators can

acquire content, and the content is protected during transport to the license creator, then all other security requirements can be delegated to the license creator. Below, this is succinctly stated.

c.1 Content is only accessible by untampered components created by the official system developers, under the conditions of a valid license issued by a bona fide license creator.

This leads to the following security requirements:

c.1.1 (authorisation) Only authenticated license creators can acquire content from the content creator.
c.1.2 (secrecy) The content is protected from eavesdropping when it is communicated to another component of the system.
c.1.3 (delegated) All requirements following from l.1.

3.2 License creator

Each of the established security properties for the license creator is analysed below to arrive at security requirements. The first property was stated as:

l.1 Content is only accessible by a renderer with a valid license issued to that renderer, originating from the license creator, under the terms stated in that license.

This property is the counterpart to property c.1, and governs the fundamental protection of content in a DRM system. Content protection requires secrecy of content: the provided content must remain inaccessible for anyone and any components except the ones specifically allowed to access the content. These components are thus trusted: they may only render the contents if all terms of the accompanying license have been met. This also implies a robustness requirement: these components must be resilient enough to disruption, so that their functions (license compliance checking, etc) cannot be influenced or disrupted by the user or anyone else. To ensure that content is not leaked, the components on the user side need to be all authenticated. Hence, components may only forward content to authenticated components. As seen in the generic process model, the core protection offered by the core of DRM systems ends with rendition of the content (methods to apply protective measures to analogue content are not within the core scope of DRM systems). This constraint is also an explicit security requirement: protection ends with rendition.

Together, these requirements are sufficient according to our analysis to provide content protection in the generic process model.

The next security property for the license creator was formulated as:

l.2 Precisely deliver what has been requested, in a consumable form, at the desired time for the licensee.

This property governs the communication between a license creator and a user. The communication channel must ensure integrity of communication, and the service must be available without undue delay.

The third property was stated as:

l.3 The impact of breaking the system must be constrained.

It is not easy to qualify this exactly, as it deals with unknown attacks. Analysing this from a defensive stance, we can formulate two requirements preventing escalation when the system is broken. First, a break in the system must be local to one implementation of the system (and not applicable to all implementations); secondly, the system must offer the possibility to be updated to incorporate new protective measures.

The first requirement constrains the impact of a break of the system, while the second allows the system to respond adequately to such a break. Hence, together these requirements satisfy the desired property.

The final security property for the license creator was expressed as:

l.4 Safe and secure negotiations with the user: other parties cannot influence (hack/break) the license creator's side of the negotiations.

Examining this from a defensive stance, this property is safeguarded when delivery only occurs in exchange for compensation, when the user agrees to the terms of this exchange and when the user cannot deny this.

All the requirements for the license creator are summarized below.

l.1 Content is only accessible by a renderer with a valid license issued to that renderer, originating from the license creator, under the terms stated in that license.

Concretely, this leads to the following points:

- l.1.1 (secrecy) The content is only accessible by the manager/renderer components specified in the license on the user side and remains secret until it has been converted into an analogue form.
- l.1.2 (trust) The manager/renderer pair on the user side will only render the content if all terms of any one valid license governing this content are met.
- l.1.3 (robustness) The internal workings of the manager/renderer pair cannot be influenced or disrupted by the license creator, the user nor any third party.
- l.1.4 (secrecy) No secret information necessary for the operation of the components or pertaining to content (e.g. cryptographic keys, content) can be discerned from the manager/renderer pair, nor from the communication channel between them.
- l.1.5 (constraint) The domain in which the DRM system offers protection ends at the renderer.
- l.1.6 (authentication) no component sends the content to another component, unless the receiving component is authenticated as an official component (i.e. created by the DRM developer) and if the receiving component is allowed to receive content from the sending component.

l.2 Precisely deliver what has been requested, in a consumable form, at the desired time for the licensee.

Concretely, this leads to the following points:

- l.2.1 (integrity) The request should be received correctly.
- l.2.2 (availability) The user should be able to contact the license creator, without undue delay.
- l.2.3 (availability) The user should be able to receive content and license at any time she desires.
- l.2.4 (integrity) The content/license should be received correctly.

l.3 The impact of breaking the system must be constrained.

Concretely, this leads to the following points:

- l.3.1 (mitigation) Prevent "Break Once, Run Everywhere".
- l.3.2 (adaptation) The components of the system can be updated to implement and deal with new or altered protective measures used by the system.

l.4 Safe and secure negotiations with the user: other parties cannot influence (hack/break) the license creator's side of the negotiations.

Concretely, this leads to the following points:

- l.4.1 (integrity) Delivery only occurs to a paying party, in exchange for compensation as specified by the license creator.
- l.4.2 (correctness) The user is aware of and consents to the terms of the exchange.
- l.4.3 (non-repudiation) the user cannot deny having received a license, nor having requested the received license for the exchanged compensation.

3.3 User

Below, the security properties as desired by the user are analysed and security requirements are derived from them. Again, the generic process model was used as a base to match the stated properties against, and the method of applying attack trees – adapted to stress defensive aspects – was again applied.

The first security property as desired by the user of a DRM system was established in Section 2.2 as:

u.1 Precisely acquire a consumable form of the content that the user desires, at the moment the user desires it.

This property expresses the delivery capabilities of the system. For this property to be satisfied, the license creator must be available at all times. Furthermore, he must send what he and the user agreed upon. Additionally, any communication between license creator and user must be undisruptable and uncorruptable. This all implies that the user trusts the license creator. Hence it is important that the user is convinced that the party at the other end of the communication channel is indeed the party she trusts, i.e. the license creator. This is achieved by having the license creator authenticate himself to the user.

The next security property was characterized as:

u.2 To order licenses or content on the user's behalf requires the intensional participation of the user.

This property ensures that the system may only act on the intentions of the user, and not otherwise. Safeguarding this property in the generic model implies safeguarding the communication between user and license creator. This communication consists of two distinct parts: the negotiations (resulting in the compensation) and the delivery of content and license. To safeguard all communication, the user is required to authenticate herself. Naturally authentication data is private. Secondly, to safeguard negotiations, compensation (payment) only occurs intentionally, and the license creator cannot deny having received compensation.

The third security property was expressed as:

u.3 Neither content nor licenses can be linked to the user.

This property ensures the privacy of a user using a DRM system. The privacy property breaks down according to the generic process model: the user's privacy must be ensured by all parts interacting with the user. In the generic process model, these are the license creator, the communication channel from the license creator to the user and the components of the DRM system at the user's side, i.e. the content processor and the content renderer. The license creator must handle that personal data he receives with care, and within the limits set / agreed upon by the user; the communication channel must ensure the privacy of the user; and the components on the user side safeguard the user's privacy from third parties.

The final security property for the user was formulated as:

u.4 The user is aware of all negotiations resulting in an agreement between her and the license creator, and consents to the terms of any such agreement.

This property ensures that the system cannot cheat the user on the delivered terms. For this to hold, the terms must be unalterable in communication. This would suffice, if the license creator is trusted to send the terms as negotiated with the user. A bona fide license creator will do this, if he is ensured that compensation has been (or will be) received.

All the requirements for the user have been summarized below:

u.1 Precisely acquire a consumable form of the content that the user desires, at the moment the user desires it.

Concretely, this leads to the following points:

- u.1.1 (availability) The services of the license creator are at any time available to the user.
- u.1.2 (integrity) Communication between the content binder component and the content processor component cannot be disrupted or corrupted.
- u.1.3 (trust) The license creator sends what has been agreed upon.
- u.1.4 (authentication) The license creator must authenticate itself to the user.

u.2 To order licenses or content on the user's behalf requires the intensional participation of the user.

Concretely, this leads to the following points:

- u.2.1 (authentication) To obtain content or a license, the user must authenticate to prove its identity (or pseudonym).
- u.2.2 (secrecy) The user's authentication data is only available to the user.
- u.2.3 (compensation) Payment only occurs with the user's knowledge and consent.
- u.2.4 (non-repudiation) the license creator cannot deny having received payment, nor having received confirmation of consent to the terms.

u.3 Neither content nor licenses can be linked to the user.

Concretely, this leads to the following points:

- u.3.1 (trust, privacy, secrecy) The license creator only saves those personal data, for which the user has given permission.
- u.3.2 (trust, privacy, secrecy) The license creator only uses those saved data for purposes for which the user has given permission.
- u.3.3 (privacy, secrecy) No information concerning the user that the user desires to keep private can be learned from the communication between the user and the license creator (in particular from license/content requests and retrievals).
- u.3.4 (trust, privacy, integrity) No third party can acquire personal data concerning the user from the content processor, the renderer or communications between these two components.

u.4 The user is aware of all negotiations resulting in an agreement between her and the license creator, and consents to the terms of any such agreement.

Concretely, this leads to the following points:

- u.4.1 (integrity) No third party can alter or disrupt the communication of terms as sent by the license creator to the user.
- u.4.2 (trust) The license creator must abide by the terms the user agrees to (and pays for).
- u.4.3 (availability) The (financial) compensation of the user for the terms set by the license creator must become known in a timely fashion to the license creator.

3.4 Consequences

The security requirements established in the previous section have some consequences, which are mentioned below.

It follows from the requirements derived from the license creator's desired security properties (l.1, l.2 and l.3) that the components on the user's side should function as a trusted computing base. The points following from property u.3 indicate that the license creator should provide a privacy statement. Requirement u.1.3 implies that the license creator provides a security policy.

4 Illustration of applicability

In this section we will illustrate the use of the generic process model and derived security requirements to practical DRM systems. It is not our intention to provide a complete analysis of existing DRM systems, but to indicate the possibilities of applying our findings.

Our first case study was performed on a system from a Dutch company which develops DRM systems. This case study was based on our work, previously on reported in [11]. The system matched our model very well, and our security analysis pointed out a missed requirement in the system (requirement u.4.5). They took into consideration implementing a mechanism to assure the user non-repudiation.

The remainder of this section will first compare the results of this chapter (i.e. the generic process model and the security requirements upon it) to several high-level, generic DRM systems. This is followed by noting how to use the results in verifying DRM design. The section concludes by applying the found requirements to two popular DRM systems, *Windows Media DRM* from Microsoft and *iTunes* from Apple. These systems are interesting for a number of reasons: these systems are the most successfully deployed DRM systems in the current market, and both have been successfully attacked. Observe that both systems are refinements of the model depicted in Figure 3.

4.1 Comparison to generic DRM systems

Several DRM systems are being developed with a focus on genericity. Examples include Coral [3], which focuses on interoperability; Open Mobile Alliance (OMA) DRM [13], which is mainly geared towards mobile devices; and Marlin [2], which is aimed at providing DRM for content accessed using varying services and devices in the bounds of a domain. All three of these have a broad target, and thus were developed as generic DRM systems. Because of this, they are perfectly suited to illustrate how to relate our generic results to real DRM systems.

Coral The Coral architecture [3] is developed to act as a basic building block for DRM systems. By itself, it does not constitute a DRM system, but is intended as a core framework that enables interoperability of DRM systems. This means that content is fed to it from one DRM system, then content is processed and finally, with an appropriate set of access conditions, content is forwarded to another DRM system. As such, interoperability of DRM systems seamlessly matches our generic process model.

As Coral focuses on providing interoperability, it has a vastly expanded set of roles. Most of these roles constitute specific refinements of the license creator's role, that enable interoperability of licenses between DRM systems.

As Coral acts not as a DRM system, but an interoperability layer, the role of the content creator is not strictly necessary. This functionality is taken over by the roles of content importer and exporter.

Open Mobile Alliance DRM The OMA DRM [13] system's primary focus is to enable DRM on mobile devices, such as cellphones. In this context, super distribution and accessibility of content for devices grouped into domains is important. Like Coral, OMA DRM is developed for implementation. The design documentation for OMA DRM provides an architecture, which acts as the counterpart to the generic process model, and a requirements document [**?**] that (also) provides the counterpart to the security requirements.

The functional architecture of OMA DRM closely resembles the generic process model. It consists of a content issuer (the content creator), a rights issuer (the license creator) and a DRM agent (the user). Nonetheless, the security requirements are all aimed at protecting content while interacting with the user. The focus of the security requirements of OMA DRM is upon requirements following from properties c.1 and l.1. Other requirements, especially the user's security

requirements, are underrepresented in OMA's DRM. OMA DRM's specifications are at an implementation level, lacking a generic nature. This has caused their security requirements to focus on implementation aspects of security, and understate other requirements.

This observation does not imply that there OMA DRM suffers from a security risk – but the set of requirements used as input for designing the system omits several requirements established in Section 3.

Marlin Marlin [2] is a DRM architecture being jointly developed by several large consumer electronics companies (Sony, Samsung Philips, Panasonic) with additional input from a specialised DRM company (Intertrust). Given this strong CE manufacturer backing, it comes as no surprise that Marlin is targeted at providing DRM for CE-devices, and ensuring interoperability of DRM across all devices belonging to one user. The Marlin developer community has created Octopus, a reference core impementation of a DRM system complying to the Marlin framework. Octopus, like OMA, focuses upon the user, and thus the distinction between content creator and license creator is not made explicit.

That notwithstanding, the actors as specified in the core Marlin architecture conform to the roles in the generic process model. Marlin's focus on CE devices and portability of content amongst devices owned by the same user has led to a refinement similar to one in OMA, namely the introduction of domains. This refinement introduces new security concerns, however, as Marlin recognises the same three core roles as the generic process model, these concerns are addressed by requirements following from the stated desired security properties. Unfortunately, Marlin security requirements are not publicly available, so a comparison between them and the stated security requirements in Section 3 cannot be made.

4.2 Verifying DRM design

The list of security requirements can be used as a checklist when verifying security aspects of a design for a DRM system. The generic nature of the process model implies that it is applicable to most designs for a DRM system. Therefore, the requirements derived from it should be met by any matching system.

Additionally, the process model can be refined to model a specific DRM system in more detail. This refined model could be combined with the stated requirements in order to derive more detailed requirements for this specific DRM system.

4.3 Application to popular DRM systems

Both Microsoft and Apple have created DRM systems that enjoy a large use base. In this section, we take a closer look at these two systems using the established security requirements as a guideline.

Windows Media DRM Microsoft has developed a DRM system compatible with their *Windows Media Player*. They created two file formats that can be enhanced with DRM protection: Windows Media Audio (.WMA, for audio content) and Windows Media Video (.WMV, for mixed video and audio content).

Microsoft's DRM system has a role delivering protected content to the user. This content can only be accessed with a legitimate license. This manner of operating conforms to the process model depicted in Figure 3.

There have been two well-known successful attacks on Microsoft's DRM system (described more fully in [7]). The first attack consisted of replacing part of the renderer that is responsible for rendering audio (the audio driver). Protected music could now be captured after it had been converted to a form understandable by the audio driver, and saved to disc.

In our proposed security requirements, this attack would have been prevented by satisfying requirements l.1.1 and l.1.6. The part of the DRM system on which this attack worked, was the

same for each installation. This means that the attack was possible on all installed systems, which violates requirement l.3.1.

Microsoft has released a new version of Windows DRM that was not vulnerable to this attack. Naturally, content packaged under the new system was not accessible with the old system.

A tool called 'FreeMe' is available that attacks content protected for the new version. The creator of the tool has included a detailed analysis of the protection of protected files. The tool is able to determine the keys used to encrypt the content. This is a violation of requirement l.1.4. The manner in which the key could be retrieved was the same for each installed system, which again violates requirement l.3.1.

Microsoft has since patched their DRM system. The patch implements measures that support requirement l.3.1 and uses another method to hide the used keys. Patched systems are no longer vulnerable to this particular attack.

iTunes iTunes is Apple's online music store. Music is sold in the Advanced Audio Coding format (AAC, see [10, 9]). The audio data inside the AAC file is protected by encryption. The default license allows protected files to be played on up to five different computers, and to burn them to CD.

The manner in which iTunes operates, conforms to the model of Figure 3. Content is delivered to a license creator, which binds the content to a license. The user can then acquire the bound content and an appropriate license.

iTunes has suffered from two well-known attacks. The first is by Jon Lech Johansen and captures the decrypted, digital contents before it is converted into analogue form. This attack is similar to the first attack upon Microsoft's Windows DRM system, and the remarks made then apply in this case as well – this attack exploits the lack of compliance to requirements l.1.1, l.1.6 and l.3.1.

The second attack on iTunes bears a resemblance to the second attack on Windows DRM: keys can be learned by outsiders, which enables an attacker to compromise the encryption of protected content. The keys used by iTunes (to protect the key to the encrypted audio) are stored encrypted with a system key. It is known how to reconstruct this key[4] for the Windows platform and for Apple's portable mp3 player. This means that it is possible to remove the encryption from the audio data for these platforms. The remarks made for the second attack on MS DRM apply here as well – this attack exploits the lack of compliance to requirements l.1.4 and l.3.1.

Recently, a third attack on the system was crafted by Johanson. He created a tool (called PyMusique) that replaces the iTunes tool to acquire songs. According to the description of the method[5], the main difference with the official tool is that it skips applying DRM to the downloaded file.

The attack thus consists of replacing a component of the system by an attacker-supplied component. However, no component should send the content to another component, unless the other component identifies itself as a legitimate component. Allowing this constitutes a violation of requirement l.1.6.

5 Conclusions

Digital Rights Management systems offer a method for content creators to allow their work to be spread digitally, without loss of recompensation. Security is the foremost enabling factor of DRM systems.

This chapter used a methodical way to derive security requirements from the incentives of the parties involved with a DRM system. A stakeholder analysis identified the key roles taking part in a DRM system, and the main incentives related to these roles. A domain analysis captured the core terminology of DRM systems, from which a process model was derived.

[4] e.g., see http://www.hymn-project.org/docs/hymn-manual.html#how-hymn-works
[5] http://www.daeken.com/2004/08/24/itunes/

The key roles, their incentives and the core terminology together led to a description of each role's desired security properties for a DRM system. Taken together with the process model, these security properties allowed us to establish the core security requirements of DRM systems.

The method used was justified by validation of the result of each step to known literature and comparison to a DRM system in development by a Dutch company.

The applicability of our findings was justified by comparison to several existing DRM systems. Exploited weaknesses of both Microsoft Media DRM and Apple's iTunes could have been identified in an early state using the requirements we presented. In the case of the system in development at a Dutch company, we were able to point out requirements that had been overlooked.

This chapter takes a practical stance towards security: it presents a list of requirements to be fulfilled. Compliance with these requirements may not be straightforward – see for example the second attack on Microsoft's Media DRM, which attacked a part that at first might have seem to comply to the relevant requirements presented here. This problem can be prevented using formal verification. Compliance of the used protocols with secrecy and authentication can already be formally verified. There is ongoing research into formally expressing other requirements (e.g. privacy) for protocols. Further developments can lead to establishing a theoretical basis in which it is possible to verify that a trusted computing base complies to the requirements upon it.

The process model used for establishing the requirements can be refined to include more details for a more thorough analysis of a particular system. This would then lead to further refinement in the requirements. Such a refinement requires a loss of the generic nature and therefore would result in confining the applicability of the resulting findings to systems exhibiting specific characteristics.

References

1. H. Chang and M.J. Atallah. Protecting software code by guards. In *Security and privacy in digital rights management*, volume 2320 of *LNCS*, pages 150–175. Springer-Verlag GmbH, January 2002.
2. Marlin Developer Community. Marlin architecture overview, 2006.
3. The Coral Consortium. Coral consortium whitepaper, 2005-2006.
4. I. Cox, J. Bloom, and M. Miller. *Digital watermarking: principles & practice*. The Morgan Kaufmann Series in multimedia and information systems. Morgan Kaufmann, October 2001.
5. S. Guth. A sample DRM system. In *Digital Rights Management*, volume 2770 of *LNCS*, pages 150–161. Springer-Verlag GmbH, November 2003.
6. J. Haitsma and T. Kalker. A highly robust audio fingerprinting system. In *Proceedings of the 3rd international conference on music information retrieval*, October 2002.
7. T. Hauser and C. Wenz. DRM under attack: weaknesses in existing systems. In *Digital Rights Management: Technological, Economic, Legal and Political Aspects*, volume 2770 of *LNCS*, pages 206–223. Springer-Verlag GmbH, November 2003.
8. B. Horne, L. Matheson, C. Sheehan, and R.E. Tarjan. Dynamic self-checking techniques for improved tamper resistance. In *Security and privacy in digital rights management*, volume 2320 of *LNCS*, pages 141–159. Springer-Verlag GmbH, January 2002.
9. International Organization for Standardization (ISO). ISO/IEC 13818-7:2004 – Information technology – Generic coding of moving pictures and associated audio information – Part 7: Advanced Audio Coding (AAC).
10. International Organization for Standardization (ISO). ISO/IEC TR 13818-5:1997/Amd 1:1999 – Advanced Audio Coding (AAC).
11. H.L. Jonker. Security of Digital Rights Management systems. Master's thesis, Technische Universiteit Eindhoven, August 2004.
12. A. Kiayias and M. Yung. Breaking and repairing asymmetric public-key traitor tracing. In *Digital Rights Management*, volume 2320 of *LNCS*, pages 32–50. Springer-Verlag GmbH, November 2003.
13. Open Mobile Alliance (OMA). OMA-DRM-ARCH-V2_0-20040715-C – DRM architecture.
14. B.C. Popescu, F.L.A.J. Kamperman, B. Crispo, and A. S. Tanenbaum. A DRM security architecture for home networks. In *DRM '04: Proceedings of the 4th ACM workshop on Digital rights management*, pages 1–10. ACM Press, 2004.

15. A. Pouloudi. Aspects of the stakeholder concept and their implications for information systems development. In R.H. Sprague, editor, *Proceedings of the 32nd Hawaii International Conference on System Sciences (HICSS-32)*, Los Alamitos, CA, 1999. IEEE Computer Society Press.
16. L. Prechelt and R. Typke. An interface for melody input. *ACM transactions on computer–human interaction (TOCHI)*, 8(2):133–149, June 2001.
17. R. Prieto-Díaz. Domain analysis: An introduction. *Software Engineering Notes*, 15(2):47–54, 1990.
18. R. Safavi-Naini and Y. Wang. Traitor tracing for shortened and corrupted fingerprints. In *Digital Rights Management*, volume 2320 of *LNCS*, pages 32–50. Springer-Verlag GmbH, November 2003.
19. Bruce Schneier. *Secrets & Lies: Digital Security in a Networked World*. Wiley, 2000.
20. C. Serrão, D. Naves, T. Barker, M. Balestri, and P. Kudumakis. Open SDRM – an open and secure digital rights management solution. June 2003.
21. W. Shapiro and R. Vingralek. How to manage persistent state in DRM systems. In *Security and privacy in digital rights management*, volume 2320 of *LNCS*, pages 176–191. Springer-Verlag GmbH, January 2002.
22. I. Sommerville. *Software Engineering*. Pearson, 2004.