Core security requirements of DRM systems

Core Security Requirements of DRM Systems

H.L. Jonker and S. Mauw

Eindhoven University of Technology
Department of Mathematics and Computer Science
P.O. Box 513, NL-5600 MB Eindhoven, the Netherlands
{h.l.jonker,s.mauw}@tue.nl

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. Although security of DRM systems has received quite some attention, a systematic overview of the core security requirements for DRM is missing. In this paper we will conduct a stakeholder analysis and develop a simple, generic conceptual model to arrive at such a set of core security requirements.

1 Introduction

There has been a long and ongoing struggle to secure content (a term used to indicate works of art, such as music, literature, movies, etc.) against unlimited copying. Computers have had a strong impact on this struggle, as creating a copy on a computer is almost effortless. This has traditionally been seen as a threat to content creators. However, with the recent rise of broadband Internet connections, this also means that selling and delivering content online is becoming a viable business venue. The main obstacle for this 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 [19], traitor-tracing [11, 17], watermarking [2], fingerprinting [4, 15] and tamper resistant code [6, 1]. There have also been various proposals for models of DRM systems with specific properties [3, 13, 12, 18].

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., in the case of MOSES [18], interoperability). 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 "rip" the 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 4.2.

There seem to be several reasons why the core requirements of DRM systems have not yet been made explicit in a systematic way. Digital rights management as a research topic is still a young and developing field. As mentioned before, research into digital rights management has been focused on specific architectures that solve particular problems and on security techniques which are used in DRM systems.

This paper is a continuation of the work started in [10]. That work was based on practical observations, which led to a practically oriented overview of security aspects of DRM systems. This paper uses a more structured approach to identify core security requirements and provide a justification for them.

The goal of this paper is thus focused on a particular aspect of the design of DRM systems, viz. security. Our aim is to systematically derive the core security requirements. 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.

We would like to acknowledge the assistance of several persons. 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. Finally, we thank Yan Liu for proofreading this paper.

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. [16]), stakeholder analysis (for an overview see [14]), system decomposition (see e.g. [20]). 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.

Our problem analysis resulted in a description of DRM systems, including the various incentives for parties to participate in the use of a DRM system. Using this description, the terminology involved was established. We then focused on establishing the main stakeholders to concentrate on the core of DRM systems, which allowed us to establish the more important terms of the terminology. The list of stakeholders contained such parties as: media company, developer, user, network provider, reseller. Each of these parties can play at least one role in a DRM system.

The analysis distinguished three types of core[1] participating roles: the content creator, who creates content; the license creator, who binds the content to a license; and the user[2], who wants access to content. This list of core roles emphasised the following terms of the terminology: (acquisition of) content, (acquisition of a) license, renderer, access.

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

The content creator is a role filled by stakeholders that create content, such as media companies. They can 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 content's value for users (e.g. by including the 'making of' footage), or it could increase revenue for the content creator (e.g. by including commercials). Besides that, using DRM technology it 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.

The license creator binds content to 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 the content is bound to a license created by the license creator, most content distributed in this way will not spread beyond the license creator's control.

Users will mainly be drawn to DRM systems since they 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 could offer users, is the possibility to restrict the access to (and thus the cost of) content to precisely what the user wishes.

The considerations and reservations mentioned in the descriptions above indicate desired security properties. These considerations are translated using the terminology to establish desired security properties in Section 2.2.

2.2 Desired security properties per role

The second step of the problem analysis consists of deriving the desired *core security properties* of each relevant stakeholder, by combining the stakeholder analysis and the established terminology.

[1] They each play a crucial role – they must all be present for a DRM system to function

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

The high-level properties of this section are then made precise as *concrete security requirements* in Section 3.

For each core role of DRM systems a translation of the 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 processes modelled by the conceptual process model of Figure 1.

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 protective measures a DRM system provides for digital content.

In Table 1 the incentives 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. They would be sufficient, if the descriptions in Section 2.1 of the incentives are complete, and if the translation is correct and complete.

content creator	
m.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 for those components.
license creator	
r.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.
r.2	Precisely deliver what has been requested, in a consumable form, at the desired time for the licensee.
r.3	The impact of breaking the system must be constrained.
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 started with one component for the three core roles in a DRM system: the content creator, the license creator and the user. The content creator's component is linked to the license creator's, and the license creator's to the

user's. Components for other roles were left out as they can be introduced when necessary - such additions would merely constitute refinements of the model. The resulting model was subsequently refined to incorporate various specifics of DRM systems.

The first refinement was motivated by distinguishing two types of content: digital content (which the system is to protect), and analogue content (which a user can consume). This results in splitting the user's component in two: one to convert digital content to analogue content (the *content renderer*), and one to communicate with the license creator's component (the *content processor*).

The second refinement was instigated by a design decision common to all DRM protected content: access to the protected content is bounded to a license. This results in splitting the license creator's component in two parts: one providing the bounded content (*content binding*), and one providing accompanying licenses for said content (*license creation*).

This leads us to the conceptual model depicted in Figure 1:

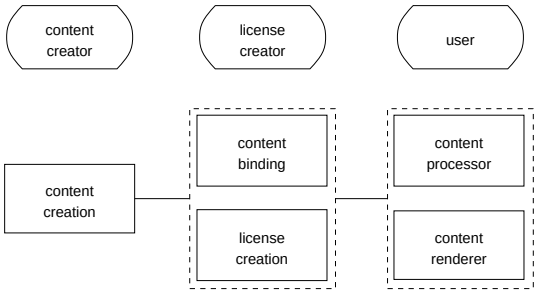

Fig. 1. A generic model of the core processes in a DRM system

The processes modeled in Figure 1 operate as follows: The content creator creates the content, which is bound to a license by a license creator. The user can acquire the bound content and the appropriate license, which the content processor uses to access the content (if and when allowed). The content is then rendered by the content renderer.

As Figure 1 models the core roles and processes in DRM systems, it can be refined to comply with almost all existing DRM architectures. For example, to comply with the functional architecture of the Open Mobile Alliance [12], the model can be refined to incorporate the roles of content issuer and license issuer.

2.4 Conclusions 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 model shown in Figure 1. In Section 2.2, they

were reworded to describe the precise desires of the stakeholders. In this section, that rewording is translated into security requirements upon the model.

Below, each requirement of the previous section is refined into security properties for the model depicted in Figure 1. To describe a specific DRM system in greater detail, both that model and these requirements could be refined.

m.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 for those components.

Concretely, this leads to the following points:

1. (authorisation) Only license creators can acquire content from the content creator.
2. (secrecy) The content is protected from eavesdropping when it is communicated to another component of the system.
3. The points discussed below at r.1 for content accessibility.

r.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:

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.
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.
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.
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.
5. (constraint) The domain in which the DRM system offers protection ends at the renderer.
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.

r.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:

1. (integrity) The request should be received correctly.
2. (availability) The user should be able to contact the license creator, without undue delay.
3. (availability) The user should be able to receive content and license at any time she desires.
4. (integrity) The content/license should be received correctly.
5. (authentication) Only authenticated users with whom an agreement has been reached will receive licenses for the content for which that agreement was reached.

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

Concretely, this leads to the following points:

1. Prevent "Break Once, Run Everywhere".
2. (updatability) The components of the system can be updated to implement and deal with new or altered protective measures used by the system.

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:

1. (availability) The services of the license creator are at any time available for the user.
2. (integrity) Communication between the content binder component and the content processor component cannot be disrupted.
3. (trust) The license creator sends what has been agreed upon.
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:

1. (authentication) To obtain content or a license, the user must authenticate to prove its identity (or pseudonym).
2. (secrecy) The user's authentication data is only available to the user.

u.3 Neither content nor licenses can be linked to the user.
Concretely, this leads to the following points:

1. (trust, privacy, secrecy) The license creator only saves those personal data, for which the user has given permission.
2. (trust, privacy, secrecy) The license creator only uses those saved data for purposes for which the user has given permission.
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).
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:

1. Payment only occurs with the user's knowledge and consent.
2. (integrity) No third party can alter or disrupt the message containing the terms as sent by the license creator to the user.
3. (trust) The license creator must abide by the terms the user agrees to (and pays for).
4. (availability) The (financial) consent of the user to the terms set by the license creator must become known in a timely fashion to the license creator.
5. (non-repudiation) the license creator cannot deny having received payment, nor having received confirmation of consent to the terms.

It follows from the requirements derived from the license creator's desired security properties (r.1, r.2 and r.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 constructed model and derived security requirements to practical DRM systems. It is not our intention to provide a complete analysis of actual DRM systems, but to indicate the implications of using our results.

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 [9]. 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 describe a comparison of 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 1.

4.1 Microsoft *Windows Media DRM*

Microsoft has developed a DRM system compatible with their *Windows Media Player*. Microsoft has 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 1.

There have been two well-known successful attacks on Microsoft's DRM system (described more fully in [5]). 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 r.1.1 and r.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 r.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 r.1.4. The manner in which the key could be retrieved was the same for each installed system, which again violates requirement r.3.1.

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

4.2 Apple *iTunes*

iTunes[3] is Apple's online music store. Music is sold in the Advanced Audio Coding format (AAC, see [8, 7]). 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 1. 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 r.1.1, r.1.6 and r.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 r.1.4 and r.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.

[3] http://www.apple.com/itunes/

[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 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 r.1.6.

4.3 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.

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 paper 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.

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 three 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.

By using a methodical way to derive security requirements for DRM systems, this paper leads to an improved understanding of the security aspects of DRM systems.

This paper 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. I. Cox, J. Bloom, and M. Miller. *Digital watermarking: principles & practice*. The Morgan Kaufmann Series in multimedia and information systems. Morgan Kaufmann, October 2001.
3. S. Guth. A sample DRM system. In *Digital Rights Management*, volume 2320 of *LNCS*, pages 32–50. Springer-Verlag GmbH, November 2003.
4. J. Haitsma and T. Kalker. A highly robust audio fingerprinting system. In *Proceedings of the 3rd international conference on music information retrieval*, October 2002.
5. 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.
6. B. Horne, L. Matheson, C. Sheehan, and R.E. Tarjan. Dynamic selc-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.
7. 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).
8. International Organization for Standardization (ISO). ISO/IEC TR 13818-5:1997/Amd 1:1999 – Advanced Audio Coding (AAC).
9. H.L. Jonker. Security of Digital Rights Management systems. Master's thesis, Technische Universiteit Eindhoven, August 2004.
10. H.L. Jonker, S. Mauw, J.H.S. Verschuren, and A.T.S.C. Schoonen. Security aspects of DRM systems. In R. Pellikaan, editor, *25th Symposium on information theory in the Benelux*, 25th Symposium on information theory in the Benelux, pages 169–176, Kerkrade, The Netherlands, June 2004.
11. 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.
12. Open Mobile Alliance (OMA). OMA-DRM-ARCH-V2_0-20040715-C – DRM architecture.
13. 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.
14. A. Pouloudi. Aspects of the stakeholder concept and their impliciations 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.
15. L. Prechelt and R. Typke. An interface for melody input. *ACM transactions on computer–human interaction (TOCHI)*, 8(2):133–149, June 2001.
16. R. Prieto-Díaz. Domain analysis: An introduction. *Software Engineering Notes*, 15(2):47–54, 1990.
17. 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.
18. C. Serrão, D. Naves, T. Barker, M. Balestri, and P. Kudumakis. Open SDRM – an open and secure digital rights management solution. june 2003.
19. 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.
20. I. Sommerville. *Software Engineering*. Pearson, 2004.