TECHNISCHE UNIVERSITEIT EINDHOVEN,
Department of Mathematics and Computer Science

MASTER'S THESIS

Security of Digital Rights Management Systems

By H.L. Jonker

Supervisor: dr. S. Mauw
Advisors: ir. J.H.S. Verschuren
ir. A.T.S.C. Schoonen

October 2004

Preface

This Master's Thesis is the end result of the final project of my study of Computer Science at the Technische Universiteit Eindhoven.

The work for this Master's Thesis has been done at TNO ITSEF bv. in Delft and at the Technische Universiteit Eindhoven. The project researches security of Digital Rights Management systems.

I am impressed with the friendly, open atmosphere at TNO. This proved to be a stimulating work environment and I am grateful that I had the opportunity to do a large part of my work there.

I would like to Sjouke Mauw, Jan Verschuren and Lex Schoonen for guiding me through this project and their constructive commentary in writing this thesis. The assistance of Cas Cremers in evaluating a security protocol and this document was invaluable, for which I am grateful. I would also extend my gratitude to Benne de Weger and Erik de Vink for taking place in the board of evaluation for this Master's Thesis.

Finally I would like to thank my family and my friends for their support during my study and during this final project.

Hugo Jonker
Eindhoven, 16 August 2004.

Summary

This Master's Thesis investigates security aspects of Digital Rights Management (DRM) systems. DRM systems are systems that enforce specific rights on digital content (e.g. music, movies, etc.). Content owners have been looking into new ways to protect their content, especially with the emergence of large peer-to-peer networks and the failure of the copy protection on DVD.

Two legal issues are intimately connected to DRM: copyright and privacy, both of which are (for most countries) regulated by international conventions.
Copyright law has always tried to maintain a balance between benefiting society (by making content public) and stimulating creation of content (by protecting content owner's rights). How DRM systems affect this balance has yet to be determined.
Privacy law recognises an individual's right to remain anonymous. Automated processing of data (such as occurs in DRM systems) threatens that anonymity and is therefore regulated. The rules to which DRM systems must adhere are mentioned.

DRM systems are client-server architectures. They work by wrapping the content in a secure container. This container cryptographically protects the content against access. To allow legitimate access, a license is needed. The license specifies access terms that must be met before access is allowed. On the user's side, a Trusted Computing Base is required to ensure the security of the system. Various security techniques are used in DRM systems. Identification techniques include watermarking, fingerprinting and including a Digital Object Identification mark. Tracing techniques include watermarking and traitor tracing. Cryptography is used for a number of purposes, including keeping the cargo of the secure container secret, providing authentication and ensuring secrecy. One of the characteristics influencing DRM design is the network on which it is to be deployed.

Finally, a generic, security-conscious model of DRM systems is presented. This model is the first model of DRM systems that is generically applicable. The main reason for its creation is to clarify security considerations of DRM systems. The model has been validated in a use case study. It proved to be of valuable assistance in assessing the security of a practical DRM system. As it focuses upon the security of a system, security issues, security design decisions and security considerations are brought to the foreground and examined.

Table of Contents

1. Introduction

Acquiring digital content (such as music or movies) over the Internet has become commonplace in recent years. At first, websites and ftp servers provided the media to those few with internet connection and the know-how to find those sites. Then, Napster came along. It dramatically increased the scale of file sharing and the ease of use for the average person. Media corporations saw Napster as a threat to their revenues and took legal action against Napster. Although the Napster user base did decrease, other so-called peer-to-peer programs filled the gap left by Napster. Nowadays, there exist many peer-to-peer networks that cater to many end users. A test run found five e-Donkey servers with more than 100,000 users, one of which offered in excess of 67 million files. There were many more servers whose figures were not this high. In another test run, Kazaa offered more than 10 terabytes of data available for download. A site hosting bittorrents [1] of Japanese cartoons showed a total data transfer in excess of 175 terabyte. These incredible figures indicate the size of the peer-to-peer community.

After years of legal battles, Napster has started again, this time with contracts with major record labels. This event marks the changing attitude of content providers (companies who own media). Companies are looking into ways to sell their content (music, movies, etc.) over the Internet. However, these companies have learned their lesson: they wish to sell their content without the buyer being able to further distribute the work, so some form of copy protection is required.
There exist copy protection problems in other areas as well: CD's are frequently copied, and the copy protection scheme for protecting DVD's has been broken (the code to do so, DeCSS, is widely spread over the internet in various incarnations. One of the more amusing ones is found on [2]).

These cases have something in common: content providers want to control under what circumstances buyers have access to the acquired media. In short, content providers desire access control. They can use this as copy protection. This control can be implemented most easily by digital means. A system that implements such measures is called a Digital Rights Management (DRM) system.

One might wonder if the public will ever adopt such a system for acquiring content over the Internet. In recent times, two important changes have occurred which makes this more likely. People have become accustomed to acquiring content over the Internet, and they are becoming accustomed to paying for transactions over the Internet (e.g. online auctions). Apple's iTunes® has sold 14 million songs in the first seven months of its existence [3]. This strongly indicates that there is a market for DRM systems.

It is important to note that DRM systems are not exclusively aimed at the Internet. A DRM system can be used in any situation where digital content needs to be protected. This means that DRM can be applied to content exchanged over a cell phone network, or in consumer electronic devices as a replacement for the insufficient protection offered by DVD's protection mechanism.

All this indicates that DRM is an important emerging technique. There is a lot of corporate interest as well as interest by researchers in developments in this field. Most of the research in this field is directed towards specification, standardisation and security issues of sub problems. There is relatively little research into the security of DRM systems. However, it is clear that the notion of DRM systems was conceived to solve a security issue. This means that DRM systems have a security goal: protect digital content from unauthorised access. In order to determine how well a DRM system achieves this goal, an understanding of the way parts of the system interact and the various security concerns at a global level is needed.

The main goal of this Master Thesis is to investigate the security considerations of DRM systems. Before doing so, the context surrounding DRM systems is described, as well as a description of some of the more important technical terms and ideas used in DRM systems.

In Chapter 2, some of the legal and commercial aspects of DRM systems are described. Chapter 3 provides information on concepts and techniques which are used in DRM systems. Using this knowledge, it is possible to construct a generic model of DRM systems. This model and the security considerations that can be applied to it are described in Chapter 4. To validate this model, a case study has been done. The results of this case study are described in Chapter 5. Finally, Chapter 6 provides conclusions, future works and closing remarks.

2. Non-technical aspects of DRM systems

In this chapter, three non-technical aspects of DRM systems are described. There exist international conventions that both create a legal framework in which DRM systems are allowed and restrict their application. Therefore, some of the legal aspects of DRM systems are noted.
Furthermore, the ultimate goal of any company is to make a profit. This is also valid for companies currently researching DRM systems. The exchanges of value that are an inherent part of a commercially deployed DRM system are described in the second part of this chapter.
The last section mentions some of the collaborative organisations that seek to develop (supporting standards for) DRM systems as well as pressure groups that closely monitor DRM developments.

2.1. Legal aspects

DRM systems have to deal with two legal issues: copyright and privacy. In the first subsection, copyright issues are examined. In the second subsection a closer look is taken at privacy issues.

Both these issues are governed by international conventions. The regulations set forth by these conventions will be shortly mentioned. There will be more specific information regarding the legal situation in the Netherlands.

Please keep in mind that both issues are currently under close inspection. The laws governing them will probably be changed in the near future. Furthermore, it is wise to note the information in this section does not constitute legal advice.

2.1.1. Copyright law

Copyright law governs the right to copy works of science and arts. DRM systems are created to protect digitised versions of these works, so they need to abide by the limitations of copyright law.

As mentioned in the introduction to this section, there exists an international convention which governs copyright law for its participants. This convention is the *Berne Convention* of 1971 [4]. The convention (and thus copyright law) aims to strike a balance between stimulating innovation (by protection the fruits of innovation) and dissemination of information (by limiting the duration of the granted protection). Any protection offered by the Berne Convention thus has a finite duration. (Note: this does not imply any obligation to make content publicly available, it merely means that the legal requirement for compensation is finite.)

As this Master's Thesis is written in the Netherlands, a closer look is taken at the Dutch copyright law. The Dutch copyright law, the *Auteurswet* [5] has been revised [6] to comply with European directive 2001/29/EG [7]. These revisions will become operative by royal decree.

The Dutch Minister of Justice makes an interesting observation [8] on copyright:

> "Op grond van zowel de huidige als de voorgestelde wetgeving is het kopiëren van werken van letterkunde, wetenschap of kunst voor eigen oefening, studie of gebruik toegestaan. De Internetgebruiker die gebruik maakt van de mogelijkheden die Napster, KazaA en vergelijkbare peer-to-peer diensten bieden om werken van letterkunde, wetenschap of kunst te kopiëren voor privé-gebruik opereert over het algemeen genomen binnen de marges van het auteursrecht. Dat geldt ook wanneer een privé-kopie wordt gemaakt van een origineel dat illegaal, dat wil zeggen zonder toestemming van auteursrechthebbende, is openbaar gemaakt."

In short, the Minster states that it is within the margins of the copyright law to acquire (digital) copies from illegal sources according to the old and new versions of the law. A bit further in the same paper he writes that it is illegal to upload digital copies to unknown receivers. In the Auteurswet people are allowed to make copies of works of art, literature or science for their own use ([5, article 16b sub 1]) – this seems similar to "fair use" clauses in other copyright laws. In the Netherlands, everyone is entitled to have a copy for study or private use.

There are other issues concerning copyright: there is no legal requirement for content to be freely available after the legal protection by the Auteurswet has ended – but when this law was created, content could not be technologically protected when it was made public. DRM technology, however, enables rights holders to specify strict access restrictions. This seems to upset the balance between stimulation of innovation and dissemination of information.

Another point is that buyers of content (e.g., a book, a CD or a CD-ROM) are allowed by law to resell content or give content away. Provisions could be made in DRM systems allowing owners to resell or to give content away.

In conclusion: copyright law is created to uphold a balance. Technological improvements have (until now) always aided the ease of copying and spreading content. Therefore copyright law needed to protect content owner's right in order to strengthen the innovative side of the balance. However, DRM technology changes this balance: content can still be protected after it has been made public. It will be interesting to see how this affects the balance.

2.1.2. Privacy law

Privacy law in the Netherlands is derived from European directives such as [27]. Privacy legislation recognizes an individual's right to anonymity. Using DRM technology enables distributors to collect personal information about individuals – playback devices (which are tied to specific individuals) can be matched to payment information, content can be matched to playback information, etc. Companies can use all this information to increase their profits at the expense of their customers' privacy.

In the Netherlands, the *Wet Bescherming Persoonsgegevens* protects the privacy of people. It is illegal to acquire and process personal data unless the persons involved have unambiguously agreed with the acquisition and processing of their personal data and the personal data is required for the execution of whatever process it is used in. It is also illegal to keep personal data longer than necessary.
This means that DRM systems are only allowed to request personal information when binding content to a specific individual. After the binding, the personal data must be removed from the systems.

2.2. Commercial aspects

The development of DRM systems costs money. This section identifies which parties will exchange value when a DRM system is used.

The European Commission has initiated a project [9] to identify which parties fill which roles in a so-called value chain for DRM systems. This project has finished. One of its results is a business model that defines roles that take part in trading multimedia documents. Using this business model, branches of industry for which DRM systems can play a role can be identified.

The reason for creating such a model is best put in the words of the project itself:
As "there is variety of different possible trading relationships, situations and arrangements, or business models, that could provide the conceptual framework for a functional specification for [DRM systems], it is important to develop a business model whose abstraction level allows to design trading scenarios with concrete transactions for the design of [DRM systems] as one of various possible scenarios" [9].

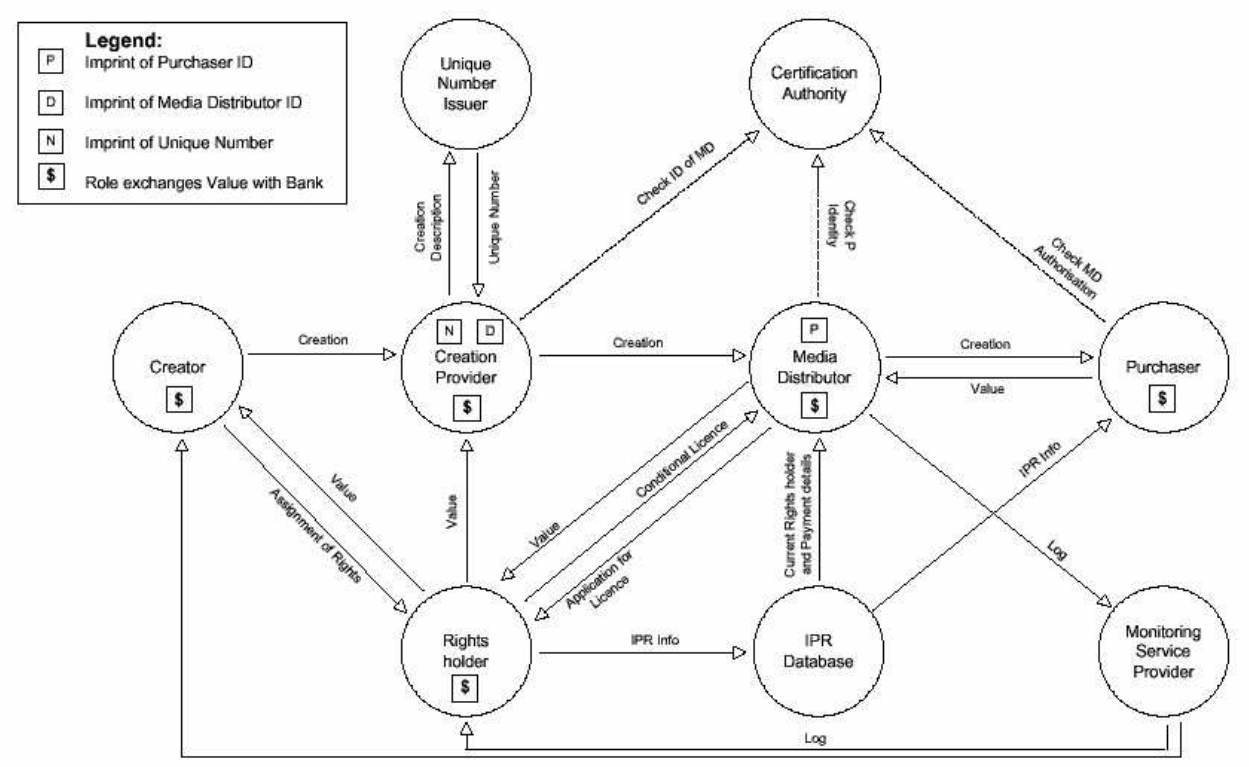

Figure 1 – The Imprimatur business model for trading multimedia documents – see [9]

In Figure 1, roles that exchange value in the process of handling content are marked with '$'. In a DRM system, companies and individuals filling these roles would expect monetary compensation for their efforts and to have to pay for services they use.

The following table matches branches of industry to value-exchanging roles (the ones marked with '$') in Figure 1:

Role	Typical branch
Creator	(groups of) individuals (artists, musicians, writers, etc.)
Creation Provider	Publishers, record companies, agencies, producers
Rights Holder	Media Corporations (record companies, movie companies, etc.)
Media Distributor	• Information / content providers (museums, libraries) • Multimedia companies / publishers • Network service providers
Purchaser	• Organisations (libraries, schools, government) • Business purchasers • Individuals

These are the companies, organisations and individuals that would have a monetary interest in a DRM system.

Not mentioned in the above model are infrastructure providers. They will charge for use of their network. Also not mentioned are IT companies. The creator of a functioning DRM system (an IT company) can sell it to the parties in the content value chain. Currently, IT companies as well as infrastructure providers (especially for the cell phone network) are publicly active in DRM development alliances.

2.3. Interested parties

Several organisations are active in the field of DRM systems. The best-known of those are named in this section to provide further sources of information.

There are several influential groups interested in DRM for non-commercial reasons. There are numerous pressure groups concerned with privacy, or against markets held by a few conglomerates.

Some of the pressure groups interested in DRM technology are:

- Free Software Foundation (FSF, www.fsf.org)
- Bits Of Freedom (BoF, www.bof.nl)
- Electronic Privacy Information Centre (EPIC, www.epic.org)
- Electronic Frontier Foundation (EFF, www.eff.org)
- American Civil Liberties Union (ACLU, www.aclu.org)
- European Digital Rights (EDRi, www.edri.org)

These organisations provide more information about the legal aspects and impact on society of DRM technology.

Aside from those groups, there are many collaboration efforts and industry alliances to create standards for DRM systems in hopes that these standards will promote interoperability between different systems. These organisations are currently the ones driving the development of DRM.

Some of the collaboration organisations of the industry involved in (aspects of) DRM technology are:

- Open Mobile Alliance (OMA, www.openmobilealliance.org)
- Content Reference Forum (CRF, www.crforum.org)
- Open eBook Forum (OeBF, www.oebf.org)
- Organisation for the Advancement of Structured Information Standards (OASIS, www.oasis-open.org)
- TV Anytime Forum (www.tv-anytime.org)
- Society of Motion Picture and Television Engineers (SMPTE, www.smpte.org)
- Internet Streaming Media Alliance (ISMA, www.isma.tv)
- Motion Pictures Expert Group (MPEG, currently residing at www.chiariglione.org/mpeg/)
- Secure Digital Music Initiative (SDMI, www.sdmi.org)

3. Technical breakdown of DRM systems

This chapter will describe DRM technology. It is an elaborate version of the work presented in [10]. As content protected by DRM systems is usually distributed in a matter similar to a client-server architecture (meaning there are a few central locations where people go, to purchase content), DRM systems are also constructed in a client-server architecture. DRM systems have to take into account the "greatest common divisor" of the devices attached to the network – or risk excluding potential customers. As these devices (and their capabilities) vary from network to network, DRM systems will also vary from network to network.

Still, there are several components and concepts that remain common to DRM systems. These are described in the first section. The second section describes the application of security techniques and technologies in DRM systems. The network specific aspects of three types of network are discussed in the third section.

3.1. Commonalities of DRM systems

Any DRM system can be divided into two sides: the distributor's side and the user's side. Generally speaking, it is far easier to implement stringent security measurements on the distributor's side than on the user's side. Therefore, this side can be made "secure", whereas the user's side can be considered "insecure". A network serves as a communication medium between the distributor and the user. For purposes of security, the network can be considered insecure.

This section first examines the distributor's side, which concerns itself with secure containers, licenses, RELs, RDDs and metadata. The next subsection describes the user's side.

3.1.1. Distributor's side

The main concern for the distributor's side is that *content* (e.g. movies, music, books) should remain inaccessible (for all) unless specific access terms are met (and then it should only be available for those people / devices specified in the access terms). To assure that the content is inaccessible, the digital content is packaged into a so-called *secure container* (see e.g. [11, part V]). The cargo of a secure container is cryptographically protected from access without the proper key. Since secure containers are protected from access, their distribution is not a security issue. This means that secure containers could even be distributed over peer-to-peer networks or copied for friends.

Access terms (*licenses*) can also be stored inside the container (so that the user acquires both content and license at the same time) or the license can be provided by other means. If the authenticity of the license is not implied by its inclusion in the secure container, the DRM system must ensure that the provided license is authentic, i.e. not forged. Licenses are typically bound, so that copying they only work on one specific device (more on this later).

To state precisely what is and what is not allowed, so-called Rights Expression Languages (RELs) have been developed (see e.g. [12]). These RELs provide the syntax for expressing

licenses. Usually, these RELs only allow the specific rights granted – any and all other rights are not allowed by the DRM.
To ensure interoperability between different DRM systems, most RELs are XML-based. In addition, a REL can have its semantics defined in a Rights Data Dictionary (RDD) (see e.g. [12, part III.2]).

Content packaged into a secure container is no longer accessible. To enable users to find the desired content, there must be a component that offers data describing the digital content. This data is called metadata. A popular choice of metadata structure seems to be the <in*d*ecs> system [13].

The distributor's side needs, at the very least, to perform three types of communication with the user: the user must be authenticated, the digital good is sent and the license is sent. This can be implemented in a myriad of ways: sending everything in one communication burst from one server; using three separate servers, one for each type of communication. If more communication with the user occurs, more communication options become possible.

A final common component at the distributor's side is a component which provides the latest version of the DRM software. This enables on the fly updating.

3.1.2. User's side

In almost all cases, the content provider who uses a DRM system requires a secure environment at the user's side. (This is not a requirement when protection of the digital data sent to the user is not desired – e.g. for low-quality content or short samples of content.) The content provider wishes to execute DRM components on the user's side. These components consist of code, data and state. Since the user's side is considered hostile territory, these components execute in hostile environment. Measures must be taken to insure validity of code execution, data integrity, state integrity and confidentiality of secret information. A tamperproof environment that cannot be inspected can provide this.

This environment is referred to as a Trusted Computing Base (TCB). A TCB functions as a trusted third party for computing. The "trust" in TCB has to do with this (and not with any trust a user would place in this component). It needs to be tamperproof to assure the correctness of calculations performed by the TCB. It needs to hide its calculations from inspection, to keep the content protected (otherwise the decryption process could be copied and then applied to the secure container outside of the TCB, which would leave the content unprotected).

A TCB provides trust for several operations requiring trust from the content provider:

- Enforcing access rights
- Opening the secure container
- Prevent licenses from unauthorised parties to be accepted as genuine

A secure TCB includes all steps from opening the secure container up to (but excluding) conversion into an analogue format. To guarantee that the terms under which the user is allowed to access the content are met, a trusted component (such as the TCB) is needed on the user's side. A DRM system that allows access to a digital version of the content is obviously flawed as this digital version can be freely copied – defeating the purpose of the DRM system. (Note that this does not mean that digital content cannot be marketed without protective measures – however, this is beyond the scope of a DRM system.)

A TCB is not needed on the user's side, if the DRM system only sends versions of the digital content to the user, which need not (or cannot) be protected (e.g. analogue versions or low-quality versions).

3.2. Security techniques used in DRM systems

DRM systems generally make use of (at least) two security techniques: cryptography to protect the content from illicit access and techniques to identify content. Identification techniques can be used to detect illicit copies and also to link DRM-protected content to a seller of said DRM-protected content. This would enable users to easily find (sellers of) legitimate copies of DRM-protected content.
In addition, tracing techniques (a specialised form of identification techniques) can be used to link illegally available copies to the legitimate owner (the prime suspect concerning who has made the copies available). Last, an important aspect to enhance the security of DRM systems is the ability to update. These techniques are discussed in the rest of this section.

3.2.1. Cryptography

The main security concern (content should remain inaccessible unless the access terms are met) implies that eavesdroppers listening in on communication between the distributor and a user should not be able to acquire a secure container and a valid license. This can be prevented by using cryptography to secure the communication channel between distributor and user. There are three cryptographic aspects to secure communications: the distributor needs to know with whom he is communicating (authentication – irrefutable proof of identity), the user wants to know if what she received is what the distributor sent (message integrity) and vice versa, and finally the actual encryption/decryption of the communication. Authentication can be done using public key cryptography, message integrity by using hashes and there are several encryption methods available. An introduction to all three these topics can be found in [14].

As stated in Section 3.1.1, a secure container protects its cargo by using cryptography. There are two ways to encrypt the secure container: using public-key cryptography or using symmetric key cryptography. Using public-key cryptography for this means the decryption takes too long for most use cases. However, symmetric key cryptography has the disadvantage of requiring the key to be distributed to the client-system in a safe and secure fashion.

A hybrid solution (transporting the symmetric key over a channel secured with public-key) is commonly used. Public-key cryptography is well suited for safe and secure transporting of small secrets. The private key can be hidden inside the TCB and the DRM system can ensure that only authorised playback devices are allowed to access the content. The speed of symmetric decryption means that the content can be streamed from the secure container. This means that there only needs to be (a small part of an) unencrypted version available whilst viewing the content. This (part of an) unencrypted version could reside inside the TCB, so it can still be protected.

A last possible application of cryptography is to protect anonymity, e.g. by using anonymous e-cash. More information on this can also be found in [14].

3.2.2. Identification techniques

An aspect of many DRM systems is that content can be identified if encountered in unprotected form. Content owners can use this (for example) to detect theft or to prove ownership, whilst users could use this to find content which they have sampled but which they have not yet acquired (e.g. they heard a friend's version, and wish to acquire that for themselves).

DRM systems can employ a variety of techniques to identify content: the digital content can be fingerprinted, a watermark can be added or a Digital Object Id can be added. All these identification techniques can be thwarted in one way or another.

Digital Object ID
The Digital Object Identification (DOI) scheme provides a lookup service – given a cryptic identifier, a server looks up the current location of the content and redirects you there [15]. This cryptic identifier is part of the metadata of the content. As this identifier is easily separated from the content, the DOI scheme is unsuited for proving ownership or detecting theft – the scheme lends itself mostly for enabling users to find content.

Fingerprinting
Fingerprinting identification (see e.g. [16]) works by matching a small sample of digital content to the original content by using a database of "fingerprints" of digital content (much the same as fingerprints are used in police investigations). Fingerprinting does not add data to the content, but uses the existing content. This means that fingerprinting can be applied to already published content.
To "take" a fingerprint, a method to do so is needed. Where the police would use ink, DRM systems could use cameras and microphones. This means that an extra device is needed for fingerprinting, which is a disadvantage. On the other hand, fingerprinting has the advantage that it can be applied to all content. All that is necessary is that a fingerprint is created and stored for that content. This makes it robust against attempts to thwart fingerprinting.
Fingerprinting techniques are currently an active research topic. There already exist some prototypes, e.g. [17].

Watermarking
Identification by watermarking (see e.g. [18]) works by embedding information in the content. They are not perceptible by humans on playback of the content (if correctly embedded), but a watermark detector finds the watermark. Watermarks can be embedded with user-specific information and thus could be able to distinguish users of the same original content. They can also embed information uniquely distinguishing proprietary content from non-proprietary content. Both applications can be made robust enough to survive the conversion from digital to analogue format.

Comparison: fingerprinting vs. DOI
DOI and fingerprinting provide similar services: a small piece of information is linked to a possibly large digital file. The advantage of DOI is that it is exact – the correct DOI always identifies the content. A fingerprint might not match and thus not identify the content, or the fingerprint database can refer (by malign intent or accidentally) to another creation resembling the fingerprinted content, but with a different rights holder. This last risk need not occur with DOI, as the content owner includes only a DOI that correctly refers to his content.
On the other hand, the DOI needs to be supplied with the content. Fingerprinting can be done at any time, even for content that existed before fingerprinting was developed. Another advantage of fingerprinting is that it can possibly be applied in much more exotic settings (e.g. "query-by-humming", in which the user hums a tune which she wishes to hear, is one such setting [17]).

Comparison: fingerprinting vs. watermarking
Watermarking and fingerprinting are complementary techniques: a fingerprint is taken from that portion of the digital good, which is perceptible for humans, whilst a watermark is embedded so that it is not perceptible. Fingerprints can only identify that content (strongly) resembles previously

fingerprinted content. Watermarks, however, can be embedded with user-specific information and thus could be able to distinguish users of the same original content. On top of that they can uniquely identify proprietary content from non-proprietary content. Both techniques can be made robust enough to survive the conversion from digital to analogue format.

3.2.3. Tracing techniques

A DRM system can use two techniques in order to trace unprotected content back to the user who originally bought a protected version of the content: watermarking and traitor tracing.

Watermarks can be used to tie the content to a specific device or user (e.g. embed something like “authorised for device 20384-1234-5678”). Alternatively watermarks can embed ownership information (e.g. “© Walt Disney”), which can be used as proof of ownership.

Traitor tracing methods have a specific application and are only mentioned here for completeness sake. Traitor tracing assumes an illegitimate device connected to the network is found, which uses a key derived from one or more legitimate keys. By using traitor tracing techniques, the legitimate keys can be retrieved from this key and thus the users whose keys were used can be identified (see [19]). Traitor tracing is mainly applicable to broadcasting schemes in which the digital content is encrypted and broadcasted over an insecure network.

3.2.4. Updatability

In order for a DRM system to cope with discovered weaknesses, it should be able to update client- and server software without affecting content that has been released under an older version of the software. This is illustrated by the fiasco of the DVD protection mechanism, the Content Scambling System. Since it does not incorporate a way to update the software, all DVD’s can currently be easily copied.

3.3. Network specific aspects of DRM systems

In this section, three types of networks are considered: cable TV, mobile communications (cellular phones) and the Internet. The specific end devices considered are decoders, cell phones and computers (as these are the most powerful devices connected to the Internet) respectively.
As the usefulness of identification techniques depends somewhat on the type of network, the possible uses are noted per network.

3.3.1. Cable TV networks

DRM systems for cable television networks usually have a decoder between the cable and the television. Since a trusted company can manufacture this decoder, this is the perfect place for implementing the TCB.

Most proprietary cable TV (e.g. Canal+) works by plugging in the cable into a decoder, which decrypts the encrypted signal and sends out an analogue variant. Digital output might be possible, but could be captured and copied. Therefore, digital output is not an acceptable solution. Normal usage of proprietary cable TV would be as demonstrated in Figure 2.

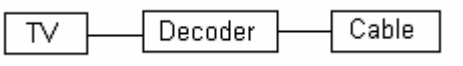

Figure 2 – Proprietary cable TV

As long as this broadcasting-like model is used, there is no real need to specify access terms. After all, the digital content is inaccessible. The user might tape the analogue output – there is little that can be done against this. See Figure 3.

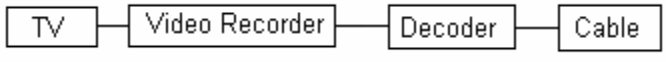

Figure 3 – Simple attack on proprietary cable TV

3.3.2. Cell phone network

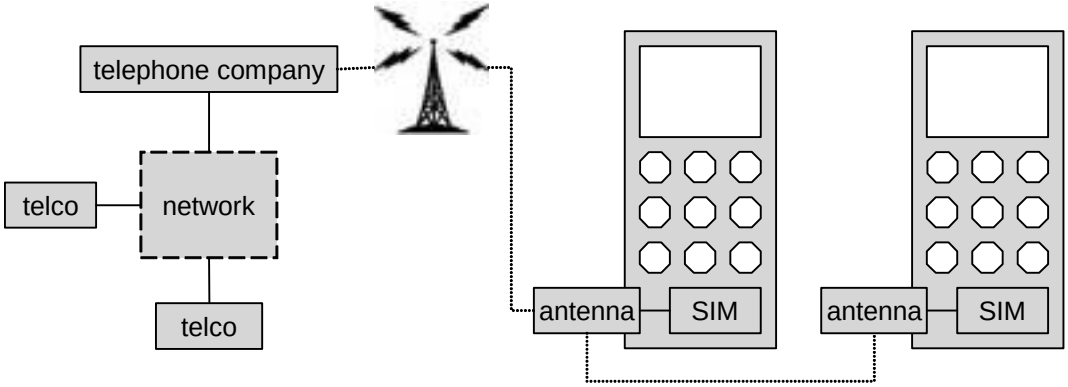

Figure 4 – Cell phone network

Figure 4 represents a typical cell phone network. Cellular phones have a short life cycle. Take for example a statement from the online magazine Wireless Web [20]:

> "Recently the GSM phone became a fashion statement and the handset replacement cycle, which translates to product lifetime, is now reduced to a year or less."

This means that when a TCB is built into cell phones, there will quickly be large numbers of potential customers with the correct hardware. Add to this hardware secrets (the SIM-card) and the closed aspect of the hard- and software (in stark contrast to computers), and it is clear that the technological requirements of DRM systems are more easily provided by the mobile phones industry (when compared to the PC market).

Currently, cell phones with extra wireless communication ports (infrared, BlueTooth) are becoming the new norm. This is also visible in Figure 4. This can be used for superdistribution of the secure containers, which means that users would exchange the secure container between their cell phones over these communication ports. The network is then only needed to supply the license.

3.3.3. Internet

It is hard to realise a TCB on personal computers without additional, tamperproof hardware components. This can be shown by the following argument: It is an accepted conjecture that computers are the computational equals of Turing machines. Turing machines can implement Turing machines, so computers can emulate computers. If a legitimate user at any one time has access to the digital source, this user can create the exact same circumstances on his computer – in other words: emulate the original state of the computer. So the user can access the digital content at any time he chooses – whilst a license could specify he is only allowed to access the content a limited number of times, or within a limited time interval.

So, current computers are theoretically insecure environments for a TCB. However, DRM is concerned with practical security: making it too hard to acquire the digital contents illegally. As long as the above theoretical attack is too difficult (or time-consuming) to execute, then the security of the DRM system can be acceptable for content-providers.

Using hardware cryptographic devices can prevent the above attack. There are two main initiatives to develop a TCB on the pc: the Trusted Computing Group (TCG) [21] and Microsoft's Next Generation Secure Computing Base (NGSCB) [22]. Both are creating a hard- and software design that implements a TCB (e.g. in personal computers). The (software part of the) NGSCB will be incorporated into a future version of Windows™.

On the Internet it is also possible to use superdistribution to distribute the secure containers.

3.3.4. Conclusion

DRM on cable TV networks sounds a bit like overkill. The technology currently in use for this niche fulfils the copy protection role adequately.

The case is different for the cell phone market. It seems that mobile networks offer the most promising platform for DRM systems – the hardware is relatively inaccessible, quickly updateable and comes equipped with a secret key. Furthermore the most avid users of cell phones are the target demographic for music and movies – both of which are prime candidates for digital distribution through DRM systems. And paying for special services on mobile phones is already a booming business (e.g. logo's and ring tones). However, their limited resources (both in storage and rendering capabilities) provide a hard limit on the content offered.

DRM systems on computers are currently viewed as a promising copy-protection mechanism and therefore there is much interest in them. However, as long as content will be offered in other forms (CD's, DVD's, etc.), these other forms can be used to convert the content to an unprotected format that lends itself for downloading.

4. A generic model of DRM systems

In this chapter, a new generic security model of DRM systems will be presented. Included are security considerations for various parts of the model. The information in the previous chapter served as a guideline in constructing the model.

The model was constructed to aid in designing DRM systems and analysing the security of DRM systems. During the research, no generic, basic model of DRM systems was found. A plethora of models with diverse security considerations was found, however, these were too specific in nature to be applicable to other DRM system models.

This model breaks new ground by being a small, generic model whose security considerations are widely applicable. To be able to coherently and consistently analyse the security aspects of this model, the security goal must be stated and a threat model must be specified.

The main security goal of a DRM system is to protect content against all access unless specifically allowed by a valid and legitimately possessed license. Secondary goals can be the ability to detect whether versions of the content are illegitimate, the ability to trace illicit content back to the perpetrator and the ability to prove authorship of content.

The threat model grants powers to a fictitious attacker. Using this model, it is possible to investigate what damage can be done with these powers to the DRM model. In the analysis below, it is assumed that a breach of security will quickly be shared amongst all users (that are connected to the network) who wish to obtain it. Therefore, the attacker is powerful: the attacker can crack weak cryptographic systems and weak keys, knows the communication protocol, controls the network, can hack (can break into systems with security flaws and alter software – e.g. the player – to create security flaws), and has complete control over the user-side device.
This is a severe threat model. Actual DRM systems can be deployed in more favourable conditions (e.g. users normally do not have complete control over their mobile phones).

In the first section, the inclusion of several components will be substantiated and reasoning for excluding other components will be given. The second section shows the model and describes security considerations for this model.

4.1. Substantiation of the model

To be able to speak of the security of DRM systems in general terms, a generic model is needed. This generic model must be adequately detailed for our needs – this means that this model must detail how the content is handled. To keep the model simple, the model is limited in scope and only details the data-flow from content-provider to an analogue rendering of the content. The substantiation covers the distributor's side and the user's side. These sides are connected through a network (of non-specific nature type).

Several models of DRM systems were used to arrive at this generic design. Most notable among these are (the architectures/models of) Open SDRM ([23]) and Moses ([24]).

4.1.1. Distributor's side of the model

Essential to the security of a DRM system is the secure container. Since the secure container cannot be accessed without a license, both the container and the license need to be incorporated into the model.

In the previous chapter the remark was made, that it was relatively easy to implement stringent security measures on the distributor's side (due to a large incentive to cooperate). Therefore, the only security sensitive points with which the model needs to concern itself are those, where the content or the license is transmitted or stored. Any other interacting components (such as those described in Section 3.1) are beyond the scope of this model. The scope is limited to keep the model small and to be as generic as possible, so that extensions can be easily added to the model.

On the distributor's side, there is a justification for four components within this scope:

- The content provider – provides the raw content
- The packager – wraps the raw content into a secure container
- The secure container
- The accompanying license

Within the scope, there is no need for a metadata component (see Section 3.1.1) or a user interface component (such as the one in Figure 7). As such, these are not included in the model.

4.1.2. The network

The communication medium – the network – is usually considered hostile. This means that both the license and the secure container must traverse hostile grounds before arriving at the user's side. As a license allows access to the secure container, it must contain some way to unlock the container – it must contain a cryptographic key. That key must be protected against eavesdropping. Whether to do this for the rest of the license or not is a design decision depending amongst other concerns upon the level of privacy desired, as is the decision whether to combine the license and the secure container or to send them separately.

In the context of superdistribution, it is logical to make the secure container freely available. To access the secure container, a user needs to acquire a license, which then constitutes to a sale of the content.

4.1.3. User's side of the model

The user's side is considered a hostile environment. There are several components at the user's side that will handle the secure container and the license. Notably absent is the operating system. The justification for this is that the operating system regulates the other components. If the security of the other components (especially the TCB) meets the requirements, then the operating system must meet the requirements. Therefore there is no reason to separately consider the operating system.

The components that can be considered on the user's side are:

- Network interface – necessary to acquire the content (e.g. wireless network card (PC), cell phone connection (GSM) etc.)
- Storage – necessary to store license and secure container if not immediately streamed (hard disk, memory card, RAM, floppy, etc.)

- Player – to allow users to access the content. Part of the DRM system.
- Trusted Computing Base – to access the secure container
- Audio / Video driver – to convert the audio / video stream into a format understood by hardware
- Audio / video card – to convert the stream into suitable signal
- Audio / video output device – to render the signal to a usable format for users (e.g. monitor / headphones, beamer /. speakers).

These terms are intentionally kept generic. This allows the model to describe more situations. This means that digital devices (e.g. monitors with a Digital Video Input port, digital amplifiers) are still covered – in the end they all convert to an analogue format (since people only have analogue inputs – by design – this cannot be avoided).

The processor is also not modelled, because the processor (like the operating system) is needed by the other components and thus the correct operation of the other components means that the processor must function as required.

The distinction between the player and the Trusted Computing Base is a narrow one. The exact nature of this distinction is a design decision: where ends the Player and begins the Trusted Computing Base? Care must be taken that unprotected content does not leak out of the Player. This might seem to put it on par with the TCB. However, for practical purposes a system might fulfil its security goals without harsh guarantees on the inability to acquire the content by leakage from the player. Evidently, such a system has less stringent security goals than others that do require the player to be secure in this respect.

4.2. The model and security considerations of DRM systems

This section first describes the generic model. An extension to the model is also shown, to illustrate this possibility. In the next subsection, security considerations for each component and each communication channel are examined.

4.2.1. The generic model

In Figure 5, the model is depicted (please note: not all security requirements are represented in this figure). The components which were justified for inclusion in Section 4.1 form the base for the model. This base is extended with communication channels, represented by lines. Some security considerations have been represented graphically: dotted lines represent secure communication channels (which means secrecy and authentication), and a rhombus at the end of a communication channel indicates that that side must authenticate itself as an authorised component to the other side.

Note the difference between authentication and authorisation: authentication is verification of identification; authorisation means having certain rights – in this case, receiving the data.
A secure channel would be meaningless if the parties between whom the channel is set up can be impersonated – therefore a secure communication channel must use some form of authentication. However, in some cases this is not enough. When secret data is transmitted over the channel, it is also important to know that the receiving side is authorised to receive the data. This means that the other side can identify itself as having been granted a right to receive the data.

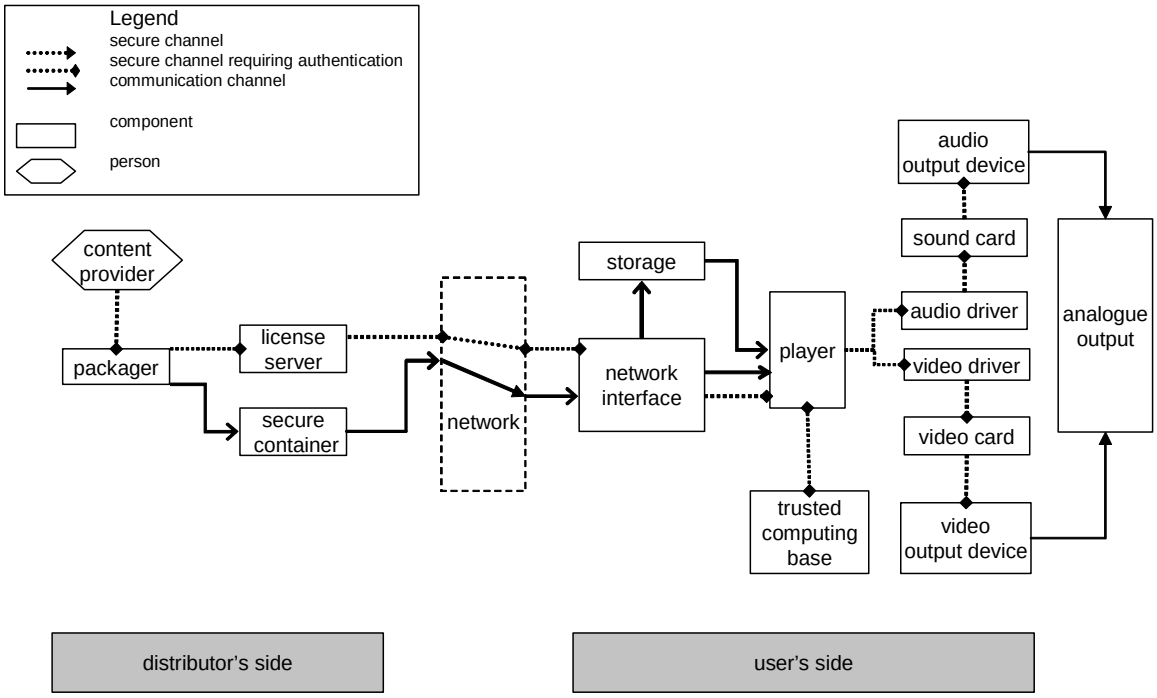

Figure 5 – the generic model of DRM systems

A careful examination of the model reveals that it is founded on two communications: the communication from the packager to the Trusted Computing Base and the communication from the Trusted Computing Base to the analogue output. However, it is not clear which component has access to what data. This can be better illustrated by layered models.

Figure 6 features a layered model of the communication between the packager and the Trusted Computing Base. As can be seen in Figure 6, the player can distinguish between the secure container and the license. However, the contents of the secure container are only accessible to the Trusted Computing Base.

To prevent modification of the secure container or the license, a secure channel is used. To ensure that the license is only sent to the correct user (in practical systems: the user who paid), an authorisation layer is used.

A similar, layered model can be constructed for the communication of content from the TCB to the user. As this is quite self-evident, such a model is not included here.

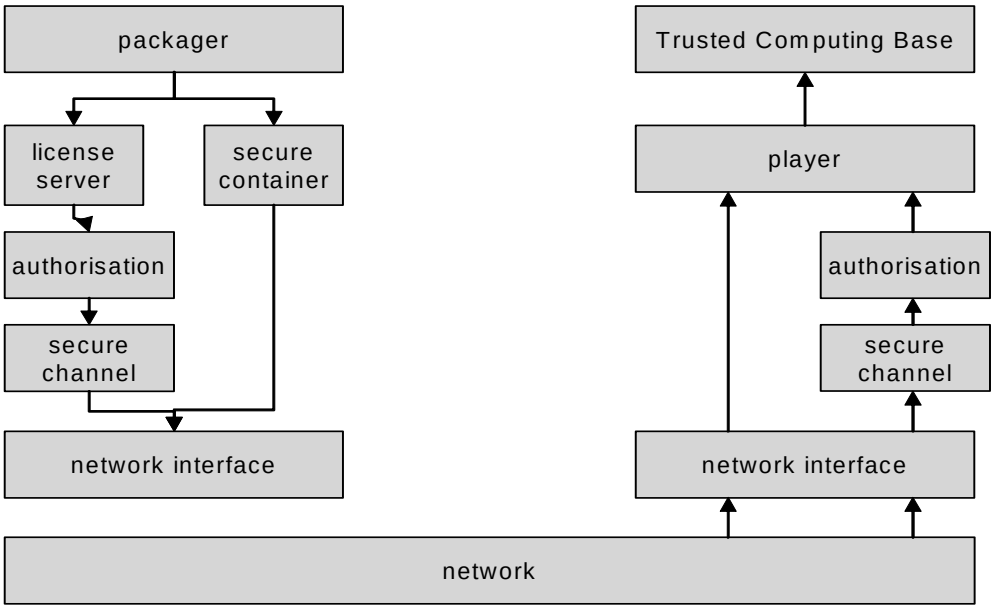

Figure 6 – Layered model of transport of content to the user side

The user is noticeably absent in this model. The reason for this is that adding a user entails making several design decisions. This clashes with the generic nature the model strives for, and so it is not present in the base model. On top of that, such an addition would convolute the model. One of the benefits of the generic model is that it is easy to grasp – this advantage would then be lost.

As stated earlier, the model can be extended to include additional components. To illustrate this, and to illustrate the convolution, Figure 7 shows a way to include a user (and his interfaces) to the generic model. The “audio output device” has been merged with “analogue out” (as have “video output device” and “analogue out” for simplicity.

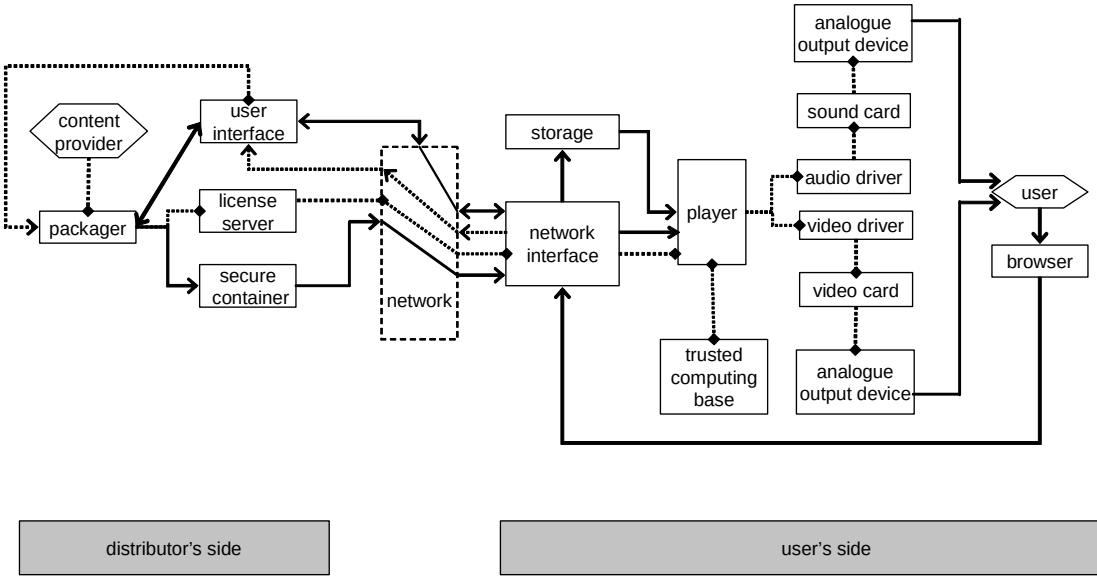

Figure 7 – adding a user to the generic model

There is no secure channel between the browser and the network interface. This is because the security of a channel between the browser and the network interface is dependant on the operating system (which is outside of the control of the DRM system). Therefore an insecure channel is depicted. The secure communication needed between the user interface on the distributor's side and the user can be secured over the network, but not inside the user's side without support from the operating system. Therefore these secure communication channels are depicted to end at the network interface.

In Figure 7 a choice has been made to create two links from the user interface to the packager. One of these links is there to determine the list of available items. The other link requests the packager to send a secure container and a license to the user.

The list of available items might also be available at the content provider. And perhaps the content provider wishes to approve all requests for content. In other words, even in this simple extension several design decisions have been made, which is detrimental to the generic nature of the model.

4.2.2. Security considerations for distributor's side

In this section, the security considerations for the distributor's side are discussed. The security considerations are given as minimal – those necessary to accomplish the security goal of the model – and additional – those that provide something extra, but are not required for accomplishing the goal.

The communication channels are listed in the order in which they occur in the communication flow. All components should adhere to certain general considerations. These considerations are mentioned in the first paragraph. For some components, additional considerations arise. These are mentioned at their proper place (according to the communication flow).

The packager has two communication channels with the player: one for the license and one for the secure container. To prevent confusion, these channels are indicated by "license and player" and "secure container and player", respectively, instead of "packager and player".

Security considerations for components
Minimal security considerations:

- Secret data may only be sent to components authorised for that specific type of data
- Secret data may only be sent over channels protected from eavesdropping

Additional security considerations:

- Components should be safe, i.e. component failure should not further damage the security of the system
- Component integrity of the receiving component could be checked before initiating communication of secret data
- Components that interact with secret data should take steps to ensure that their interactions cannot be inspected directly or indirectly by outsiders.

Security considerations for communication between content provider and packager
The minimal security considerations:

- Authentication: the packager must authenticate itself (to prevent an impersonator of the packager to receive content in the packager's stead)
- Authorisation: the content provider must prove its authorisation to add content (to prevent an impersonator of the content provider from adding bogus content to the packager)
- Secrecy: the communication channel must be secured against eavesdroppers (to prevent interception of content)

Additional security considerations:

- Non-repudiation: it may be desirable (in case of dispute) that the content provider can irrefutably prove that the packager did receive the sent content (if this is the case), and that the packager can irrefutably prove that the content was not received (if this is the case)
- Message integrity: the communication channel can be secured against corruption (to ensure that what is received is equal to what is sent)

4.2.3. Security considerations for the network

The network is considered under complete control of an attacker. Messages can be lost, arrive out of order, be replayed, altered, etc. This means that security measures are needed to communicate correctly over the network.

All components should follow the security considerations for components as described in Section 4.2.2. The license, the secure container and the communication channels are treated in the order in which they occur in the data flow.

Security considerations for the secure container
The minimal security considerations:

- The cargo of the container must be encrypted
- The container must be able to provide identifying information (so that it can be matched to the appropriate license)

Additional security considerations:

- The container could be signed to prove its origins

Security considerations for the license
The minimal security considerations:

- Signed: to prove the validity of the origin of the license, the license is signed. (note: any signing scheme provides message integrity – if it did not, the signed message could be altered, rendering the signing scheme useless)

Security considerations for communication of the license to the player
The minimal security considerations:

- Authentication (of the player)

Additional security considerations:

- Secrecy (for privacy reasons)
- Message integrity (to prevent alteration)
- Non-repudiation (either side can prove whether the communication was successful)

Security considerations for communication of the secure container to the player
There are no minimal security considerations, as the secure container prevents the content against access.

Additional security considerations:

- Message integrity

Because the secure container cannot be accessed without a license, non-repudiation, secrecy, authentication nor authorisation adds any valuable security measure.

4.2.4. Security considerations for the user's side

The user's side is under complete control of the user. This makes it as hostile as the network. Since the operations performed on this side are more sensitive than those performed over the network (i.e., accessing the secure container), extensive security measures are required.

All components should adhere to the considerations mentioned for components in Section 4.2.2. On top of that, there is the risk that components may be altered. Components should at the very least make attempts to detect this.

Neither the network interface nor the storage component is mentioned, as the security considerations they give rise to are a subset of the considerations that arise for the player. The other components and communication channels are described in the order in which they occur in the data flow.

Security considerations for the player
As this player is on the user's side, it is under complete control of the user. Any parts of the player that handle sensitive data must protect that data against attacks that can be executed due to this complete control. These protections must be strong and complete to succeed at their task. Since these parts are so strongly protected, they can be considered part of the TCB. Therefore, there are no security considerations for the player – the security of the system on the user's side is based upon the TCB.

Security considerations for communication between player and TCB
This concerns sending the license and the secure container to the TCB. Yet again, there are no security considerations. The secure container prevents the content against access and license is prevented against tampering.

Security considerations for the TCB
Minimal security considerations:

- The validity of the license must be checked (the license must come from a valid source, and the license must be intended for this TCB)
- The integrity of the license must be checked (it may not have been altered)
- The TCB must be tamperproof – this is hard to realise in practice, and therefore in practice (strong) tamper resistance is considered enough
- The TCB must be safeguarding against outside inspection – otherwise the inner process of the TCB could be determined from the outside
- Adherence to the access terms must be checked – this might mean that the TCB requires a trusted source for the current date and time. (Note: it might seem that this responsibility can be delegated to the player. However, doing so must imply that the TCB trusts the player to be as secure as it is itself (otherwise the security of the TCB is lessened). That means that the player can be viewed as a part of the TCB, and therefore this security consideration still would apply to the TCB.)

Additional security considerations:

- The TCB should be aware of its own integrity and cease to operate when this integrity is compromised.

Security considerations for communication between player and TCB
This concerns sending decrypted content from the TCB to the player.
Minimal security considerations:

- Secrecy
- Authorisation of the player to render the content

Additional security considerations:

- Component integrity of the player could be checked before the TCB initiates data transfer

Security considerations for communication between player and analogue output
The content is now decrypted. Minimal security considerations for this path:

- Each component must be authorised to receive data
- Each communication channel must be secured against eavesdroppers

Additional security considerations:

- Message integrity (to guard the integrity of the content)
- Component integrity (to be checked before sending content)

This concludes the list of security considerations applicable to the model.

4.3. Conclusions

The model presented is a small, extendable, elegant model to which security considerations have been added. This makes the model ideal for assisting in the design of a DRM system. It will also be a valuable tool when performing a security analysis of a DRM system. However, the correctness of the model with respect to real-world DRM systems still is to be proven, that is: the model still needs to be validated.

5. The generic model in practice: a case study

The model of the previous chapter was constructed on theoretical grounds. To confirm that it indeed reflects what happens in practice, a preliminary version of the model has been compared to a real world example: VirtuosoMedia's VirTunes®. Of particular interest was how the model can aid in a formal verification of a security protocol. This chapter details the findings of that exercise.

This case study was done under a nondisclosure agreement. Hence, confidential information from VirtuosoMedia cannot be disclosed. Due to this, some details in this chapter are vague on purpose. Most of these details do not pertain to DRM systems. As such, their exclusion is not detrimental to understanding the case study.

There have been two meetings with VirtuosoMedia and one informal verification session in the course of this case study. After the first meeting, a document detailing the license exchange was studied. Our commentary prompted a revised version of this document to be created. This revised version was also examined as part of the case study.

5.1. About VirtuosoMedia

VirtuosoMedia is a spin-off from the cooperation between Virtuoso Vision and Construction Media. This Dutch company develops and markets Digital Rights Management systems. As stated on their website [25]:

> VirtuosoMedia (also referred to as VM), is specialised in developing and marketing a technology that protects valuable media content. This independent, in-house developed technology offers facilities for distributing digital content (text, photographic images, music and audio-visual content) in a way that guarantees the data's copyrights. The distribution can take place over internet of on CD-ROMs.
> VirtuosoMedia is active in the field of Secure Digital Rights Management. We provide business-to-business applications.

5.2. About VirTunes

VirtuosoMedia is currently developing VirTunes®. VirTunes is a DRM system, aimed at allowing subscription holders to acquire protected digital content from online providers. The licenses required to access the content are acquired from a central point (similar to the "license server" in Figure 5). A subscription allows a user access to a fixed number of items from any subset of the available categories. These items can be accessed in a certain time period on registered devices. Of course, the number of devices that can be registered is limited.

5.3. Evaluation

The evaluation was two-folded: on the one hand, the model of Chapter 4 was verified against the product – if discrepancies were found, the model might need to be altered. On the other hand, VirtuosoMedia was interested in the security of their product. The model might point out some issues that were overlooked or had not been design considerations, but would nonetheless be of interest.

To keep this case study limited in scope, it was decided to concentrate on the communication between the license server and the player. This meant there was sufficient time to arrive at a more detailed version of the model for this aspect; besides that, protocol verification is a strong point of the security group of Faculty of Mathematics and Computing Science of the Technische Universiteit Eindhoven.
It also meant that VirtuosoMedia need not disclose all their development documents to outside scrutiny, but could limit this to a small part, that was vital in the security of the system. This agreement thus was to the benefit of both parties.

The first part of this section describes the validation of the model. The second part describes the examination of the license exchange. This second part is written in a personal style to better reflect the evaluation process.

5.3.1 Validation of the model

During the first meeting, the model of Chapter 4 was shortly discussed. VirtuosoMedia noted that currently, keeping the output secure after the player is considered by them (and other DRM developing companies) too much trouble with too little value in the overall scheme of things. Therefore, content is fervently protected up to processing in the player. After the player, this level of security is too hard to achieve without adequate support from the platform on which the player runs.

Other than that, they immediately concurred that the model was a correct base description of a generic DRM system.

5.3.2. Evaluation of VirTunes license exchange

After the first meeting, the focus was upon evaluating the security of the license exchange against the model. There were several things that were not clear to me at first. The exchange was described by a series of possible communication calls. It was not quite clear to me in which order and when these calls were to be done. Furthermore, details concerning encryption were left out as these were described in another document.

This gave rise to some interesting questions and suggestions; however these had little to do with the model or with formal verification of the protocol. In response to these questions, VirtuosoMedia created a new version of the document describing their protocol. This version included details on encryption and a sequence diagram clarifying the protocol.

In a formal verification, the protocol would have been modelled and then verified against an intruder model using e.g. Casper/FDR [26]. Unfortunately, there no longer was enough time to perform a formal verification of the protocol.

The protocol was subject of an informal verification by ir C. Cremers and myself. It seemed there was a risk of replay-attacks. Although the license would not be exposed by such an attack, it could lead to an inconsistent state, where an attempt to play content would fail. We found a remedy for this problem. Beyond that, we found a few minor noteworthy points, but no major security issues.

These results were discussed in the second meeting. The suggestion was welcomed. During this meeting, a comparison between the model and the currently described situation also came up. As the model was constructed to give insight into the security considerations and security design decisions of DRM systems, it was clear which issues can be considered for the license exchange. Most of these considerations were addressed, but not as a main goal of their design. The model helped to bring security considerations to the foreground, and to make explicit some of the implicit choices that had been taken.

The model also serves to highlight unconsidered security issues, as became apparent when discussing non-repudiation. VirTunes featured non-repudiation – the seller of content had a a designated point in the communication after which the content was considered (by design) to have been received. There was no non-repudiation mechanism for the buyer to show he had not received the content. This was not purposely so designed by VirtuosoMedia's part, it just had not been considered earlier. VirtuosoMedia is now considering this.

5.4. Conclusions of the case study

In conclusion: as noticed in this case study, before formal verification can be applied in practice, there is still quite a bit of work to do. However, as this work clarifies the issues a formal method would focus on, this leads to a better understanding of how the original product deals with these issues (and with which it does not deal at all).

Another result is that the model was applicable in this case; strengthening the belief that it correctly models the base functionality of a DRM system.
Not only is the belief that the model is correct strengthened, it has also become apparent that such a model can indeed assist in the evaluation (and construction) of a DRM system. Using the model forces the security issues, security design decisions and security considerations to be made explicit. Using the model in the design phase of a project will help to determine the exact security needs. This means that it is possible in a later phase to choose the correct security tools needed. Even if formal verification is not used, using the model for this purpose will aid the design.

6. Concluding remarks

This paper presents an introduction into DRM technology, including technological, legal and commercial aspects. Furthermore it presented the first model that focuses upon the security of DRM systems. This model has been validated in practice and found to be an accurate model. The model has been used to analyse security aspects of VirTunes® in a use case. The model has proven its worth in the use case by clarifying security considerations and bringing them to the foreground.

During the use case, it became apparent that before formal verification can take place, the item under study must comply with several requirements. Mainly among those is that the item must be explicit, exact, precise and complete about the subjects of the formal verification. This is not only necessary to perform formal verification; it also brings possibly unnoticed dualities, unintentional vagueness and other inconsistencies to the foreground.

Even without performing a formal verification, the model forces clarification of security issues, design decisions, demands and considerations. This leads to a more secure-aware design.

Because the model forces this clarification, the model is a valuable supporting tool in preparation of a formal verification of security aspects.

Bibliography

[1] Bram Cohen, *Incentives Build Robustness in BitTorrent*, May 2003

[2] http://www.everything2.com/index.pl?node=illegal+prime+number

[3] Apple Press Release, *One Million Copies of iTunes for Windows Software Downloaded in Three and a Half Days, October 20th, 2004, http://www.apple.com/pr/library/2003/oct/20itunes.html*

[4] *Berne Convention For The Protection Of Literary And Artistic Works,,* July 24th, 1971

[5] *Auteurswet 1912, September 23rd, 1912 with revisions up to July 1st, 2004*

[6] *Staatsblad van het Koninkrijk der Nederlanden*, year 2004, issue 336

[7] European Union, *RICHTLIJN 2001/29/EG VAN HET EUROPEES PARLEMENT EN DE RAAD*, May 22nd, 2001

[8] Minister of Justice J.P.H. Donner, *Nota naar aanleiding van het verslag, Kamerstuk 2002-2003, 28482, nr. 5, Tweede Kamer, p. 32*

[9] Esprit Project 20767 of the European Commission (1997), WP4, *The Imprimatur Business Model. Version 2.1*

[10] H.L. Jonker, S. Mauw, J.H.S. Verschuren, A.T.S.C. Schoonen, *Security Aspects Of DRM Systems*, 25th Symposium on Information Theory in the Benelux, pp. 169-176, June 2004

[11] Gabriele Spenger, *Authentication, Identification Techniques, and Secure Containers – Baseline Technologies*, Digital Rights Management, LNCS2770, pp. 62-80, 2003, Springer-Verlag

[12] Susanne Guth, *Rights Expression Languages*, Digital Rights Management, LNCS2770, pp. 62-80, 2003, Springer-Verlag

[13] <indecs>TM Framework Ltd., *The indecs project homepage*, http://www.indecs.org

[14] B. Schneier, *Applied Cryptography*, John Wiley & Sons, Inc., second edition, 1996

[15] Norman Paskin, *DOI detailed slide presentation*, International DOI Foundation. DOI: 10.1000/237

[16] M. Kivanç Mihçak and Ramarathnam Venkatesan, *New Iterative Geometric Methods For Robust Perepctual Image Hashing*, DRM 2001, LNCS 2320, pp. 13-21, 2002, Springer-Verlag

[17] Melodyhound.com, *Found Out What Tune Is On Your Mind*, http://www.name-this-tune.com/

[18] Fabien A.P. Petitcolas, *Digital Watermarking*, Digital Rights Management, LNCS2770, pp. 62-80, 2003, Springer-Verlag

[19] Aggelos Kiayias and Moti Yung, *Breaking and Repairing Asymmetric Public-Key Traitor Tracing*, Digital Rights Management ACM CCS-9 Workshop, LNCS 2696, pg. 32-50, Springer-Verlag, 2003

[20] Mark Paxman, *Component integration will drive evolution of 3G handsets*, http://wireless.iop.org/articles/feature/2/7/5/1

[21] http://www.trustedcomputinggroup.com

[22] http://msdn.microsoft.com/security/productinfo/ngscb/default.aspx

[23] Serrão C., Neves D., Barker T., Balestri M., Kudumakis P., *"Open SDRM – An Open And Secure Digital Rights Management Solution"*, IADIS2003, Lisbon, Portugal, June 2003

[24] European Union Project No. IST-2001-34144, *MPEG Open Security for Embedded Systems (Moses)*

[25] http://www.virtuosomedia.nl/

[26] G. Lowe, *Casper: A Compiler for the Analysis of Security Protocols*, Proceedings 10th Computer Security Foundations pp. 18-30, IEEE, 1997