

Towards a Secure and Interoperable DRM Architecture

Gelareh Taban, Alvaro A. Cárdenas and Virgil D. Gligor
Department of Electrical and Computer Engineering
University of Maryland, College Park
gtaban, acardena, gligor@umd.edu

ABSTRACT

In this paper we look at the problem of interoperability of digital rights management (DRM) systems in home networks. We introduce an intermediate module called the Domain Interoperability Manager (DIM) to efficiently deal with the problem of content and license translation across different DRM regimes. We also consider the threat model specific to interoperability systems, and introduce threats such as the cross-compliance and splicing attacks. We formalize the adversary model and define security of an interoperable DRM system with respect to this adversary. We finalize by proposing detailed protocols which achieve our security requirements. In order to achieve these requirements we provide novel applications of recently proposed proxy re-signature and proxy re-encryption algorithms.

Categories and Subject Descriptors: K.5.1 [Legal Aspects of Computing]: Hardware/Software Protection Copyrights; K.6.5 [Management of Computing and Information Systems]: Security and Protection Unauthorized access.

General Terms: Security, Design.

Keywords: DRM, interoperability, home networks.

1. INTRODUCTION

The increasing availability of computer networks, broadband technology, and the improvements in computer codec algorithms for multimedia has made the process of trading digital content through the Internet very convenient. The economic importance of online shopping for digital content, as evident with recent success stories (such as Apple's iTunes), is expected to grow even further with the introduction of portable multimedia players, next generation of gaming consoles (which will serve as media centers), as well as the emerging market for home entertainment networking. However the attractiveness of this new form of distribution is countered by the ease with which digital content can be copied and redistributed in ways that violate the intended use of the product.

In order to take advantage of the opportunities of online content distribution while at the same time preventing illegal redistribution of the content, companies have developed digital rights management (DRM) technologies thereby enforcing licensing restrictions to limit the use of their materials. A DRM system protects the value of digital content by bounding content to a license whereby the content can only be accessed (and used) under the terms stated by the license. Some examples of existing DRM systems include Apple iTunes' Fairplay [20], Secure Digital Container (SDC) [22], Windows Media DRM [25], the Advanced Access Content System [31], the Open Mobile Alliance's (OMA) DRM scheme [21], PachyDRM [32], etc.

The emerging problem however, is that most DRM systems are neither standardized nor interoperable. In general, each provider has its own technique and model to protect digital content, with little or no regard for its interoperability with other DRM systems. As a result, consumers often find they cannot render the digital content they have purchased on the device of their choice. For instance Apple's tight control over the FairPlay DRM system, has caused many iPod users to complain about their inability to play music files bought from other online services (and therefore packaged under other DRM systems) [5]. A recent survey by INDICARE [33] showed that consumers are willing to pay a higher price for more usage rights and device interoperability. From the web users polled, 86% preferred paying 1 Euro for a song that runs on any device rather than only 50 cents for a song that runs on only one device. The findings concluded that it "that it certainly pays for digital music providers to offer flexible usage rights, sharing features, and to enable the usage of digital music on various devices" [33].

This lack of interoperability is not only a concern for end users but also for content and service providers, since confused or dissatisfied customers can cause future customers to avoid legitimate digital content providers, and therefore, slow the growth of the digital industry. Ensuring DRM services compete favorably with services offering unprotected and possibly pirated content, both in terms of features and price, is a necessary step in the path to the wide adoption of DRM, and as a consequence, to the wide adoption of legitimate digital content distribution. Furthermore with interoperable DRM architectures, content providers can potentially reach a wider audience because their content will be accessible by any compliant device or application.

However, the success of any interoperability effort depends on the coalition of content owners and providers to agree on an interoperability standard. Companies planning on

Permission to make digital or hard copies of all or part of this work for personal or classroom use is granted without fee provided that copies are not made or distributed for profit or commercial advantage and that copies bear this notice and the full citation on the first page. To copy otherwise, to republish, to post on servers or to redistribute to lists, requires prior specific permission and/or a fee.

DRM'06, October 30, 2006, Alexandria, Virginia, USA.
Copyright 2006 ACM 1-59593-555-X/06/0010 ...\$5.00.

joining the coalition, need to compare the market value they obtain by adopting the standard (such as reaching a wider audience), versus the cost of joining the coalition (such as the expenses to modify an already existing architecture in order to meet the specifications).

In this paper we focus on the case of a content (or service) provider with its own DRM solution, who is deciding whether to adopt a given interoperability architecture. This decision will depend on a number of factors such as how much change is required to the provider's own DRM system so that it complies with the interoperability specifications, and also its level of assurance that interoperability will not reduce its existing DRM security. In this paper we address these concerns by proposing a general interoperability architecture that would allow content (or service) providers to minimize the change required to their DRM technology while maintaining their desired level of security as the content is exported to different DRM regimes.

1.1 Related Work

A comprehensive analysis of DRM interoperability challenges and approaches can be found in [8]. This work proposes three different approaches for creating interoperable DRM systems: Full-format, Connected or Configuration-driven. Full-format interoperability refers to a global standard that all parties adhere to. Although this can possibly be the most convenient for consumers, defining one standard for all applications and different business models is hard. Connected interoperability refers to devices that contact an online translation service. This approach is perhaps the simplest from the perspective of the content provider. However, connected interoperability implies that the portability of the content for the consumer requires online connectivity. Essentially, every time the consumer wants to transfer content to a different device, they need to be online. An important consequence is that the provider has the ability to monitor how the user controls the content and this can be a breach of the consumer's privacy. Configuration-driven interoperability is a middle ground, where end user devices can locally translate DRM protected content by downloading appropriate tools from the content provider.

A connected interoperability approach called Networked Environment for Media Orchestration (NEMO) is also presented by [8], where DRM systems are augmented with online services to ensure the validity of policies. NEMO builds a trusted messaging scheme to provide a set of requirements such as authentication and authorization of devices and content. NEMO is used as a building block by both the Coral Consortium [30] and the Marlin Initiative [28]. The latter approaches interoperability by specifying a global standard that consumer electronic companies can follow. Coral on the other hand establishes specifications for interoperability between different rights management systems. To this end, in principle Coral allows the transcoding of the content and translation of the encrypted content and licenses between different DRM regimes. The proposed system of both Coral and Marlin however, are presented at a very high level, without a detailed analysis of the system threats.

Another connected approach is explored in [14], where the authors look at the translator requirements and some implementation options. Although the authors provide examples of specific translations, again the system is high level with no security analysis.

An experimental implementation of an online and centralized proposed interoperability framework is OPERA [17]. The authors assume that the media is packed for all supported media and DRM types, however, the usage rules are provided in a system independent way. That is, the user's license is independent of the underlying DRM system used by the rendering devices. Although this project presents a detailed interoperability system, its disadvantage is that it is completely centralized and fully connected (online). This system is also not very flexible because all DRM systems have to adhere to the OPERA DRM standard. Indeed, there is no translation between different DRM systems and so, this approach is close to a full-format approach.

Proposals for standardization of several common DRM features with the objective of facilitating interoperability have recently been proposed. In [6] the authors propose that a gradual change through intermediate levels of operability can be achieved with the introduction of standards for interfaces, and protocols. Similarly in [11] the authors identify seven subsystems which should be common to all different DRM frameworks. These are Content Service, License Service, Access Service, Tracking Service, Payment Service, Import Service and Identification Service. The authors therefore, propose that these key subsystems should be standardized gradually in order to achieve higher levels of interoperability. [15] believes that the history of data formats for the expression of digital rights and licences are showing signs of consolidation.

The DRM interoperability framework that is closest to our proposal is [9]. In this work the authors propose the use of a rights issuer module as a central device in a home entertainment system. The idea is that this module serves as a content and license translator between the devices from provider A to those of provider B . Therefore the rights issuer module helps the consumer transfer digital content packaged under an exporting DRM system (DRM_A) to another importing device that uses a different DRM system (DRM_B) with minimal or even no change to existing devices. They provide various protocols for different system configurations from connected oriented to configuration oriented. Their proposed schemes however, are very high level with no implementation details. Furthermore, their work lacks a careful analysis of the new threats associated with the introduction of an intermediary module.

1.2 Our Contributions

As evident in the current literature, there is much interest in architectures that bridge the gap between the interoperability of different DRM systems. However, these research efforts are generally very abstract, presented at a very high level with no real implementation details. This makes it difficult to analyze the security of the proposed architectures.

In this paper, we provide the first steps towards a more rigorous analysis of interoperability and its security by considering two main issues. Firstly, we describe a threat model for interoperability of DRM systems and determine the security requirements needed for interoperable DRM architectures. Then we provide a detailed implementation of a variety of interoperable DRM architectures with our two main requirements: minimizing the changes that are required for participating DRM regimes, while maintaining their stand alone security. These architectures range from providing compatibility to homogeneous DRM systems (DRM sys-

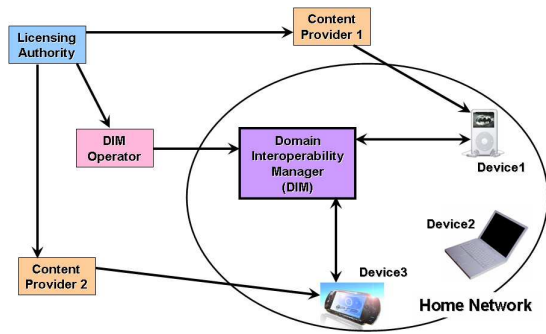


Figure 1: System Architecture

tems that use the same rights expressions languages and encryption techniques) to fully heterogeneous DRM systems. Furthermore, our proposed architectures also provide novel practical applications of the recently proposed proxy re-signature and proxy re-encryption algorithms.

2. PROPOSED INTEROPERABILITY ARCHITECTURE

We base our system in a commonly accepted model for the emerging home entertainment networking paradigm [13, 9, 28], and extend it in order to provide elements necessary for a robust interoperability framework. Our system model, shown in Figure 1, consists of the following entities:

Content Providers are organizations or companies who offer digital content to consumers protected with their own DRM tools. Protected content is bound to a set of rights, a notion that is described in a license.¹

Domain interoperability manager (DIM) Operators manufacture and sell the DIM devices that are used in home networks to manage and facilitate the transfer and translation of digital content between compliant devices, in accordance with the rights set by the content providers.

Licensing Organizations certify and manage compliant devices. This includes managing the revocation of circumvented or compromised devices. The licensing organization generally delegates individual device certification to the CE manufacturers that are contractually bound to produce only devices that are compliant with the content provider’s DRM systems.

We assume the home network domain has a domain manager (the DIM) which keeps track of the devices in the home network as well as one or more content managers (or rendering devices) that bring new data content into the domain by interacting with the content providers. Because of the introduction of the DIM as a DRM translation service, we view

¹Note that in a practical system there can be not only content providers, but service providers, DRM tool providers and consumer electronics (CE) operators. However we have assumed that the content providers perform all these tasks (service, CE devices and DRM tools) for simplicity of presentation and because this simplification does not affect our proposed algorithms when implemented in a more general setting.

the home network as one central domain consisting of all devices owned by the household. Device compliancy is checked based on individual device authentication and attestation.

The interoperability problem deals with an exporting device, D_A and an importing device, D_B in the consumer’s home network, with respective operators P_A and P_B . Assuming device A obtains new data content from a provider P_A , the objective is to translate and transfer the digital content, *in a secure and seamless manner*, from device D_A to device D_B .

2.1 Trust Model

For the entities in the above described system model to interact, we must also define the trust relationships between them. The trust model should be a guarantee, for any content provider that joins the interoperability framework, that the restrictions of the content will be satisfied by all participating parties.

The licensing organization acts as a certificate authority, with a well-known public key. The licensing organization certifies the CE and DIM operators using a signed certificate. The licensing organization also manages a global device revocation list which keeps an account of all compromised or circumvented devices. Detection and identification of the uncompliant devices is not the focus of this paper and therefore we leave it as a topic for future work.

The CE operators, who are contractually bound to manufacture and sell only compliant devices, can issue device certificates which are stored, along with the CE certificate by the licensing authority, in the device (both CE and DIM). Device D_A will have stored for example, a certificate from CE authorizing the public key of the device: PK_{D_A} , a certificate of the licensing organization authorizing CE as well as the public key that D_A device must trust: PK_{P_A} (i.e. the public key of the authority in DRM regime A , which we have assumed is the content provider P_A , as explained in the previous section).

To deliver the content, the content providers first authenticate the devices, ensuring that they have not been revoked and are compliant by verifying their device certificates. Further verification of the goodness of the device and that its software is an up to date version, can be done via attestation (we provide more details in Appendix ??). The content provider then delivers the DRM packaged content using secure communication channels. At the end of the transaction, device D_A stores the encrypted content $\{M\}_{CEK}$ (with content encryption key CEK), the license $\{CEK, R\}_{PK_{D_A}}$ (encrypted with the public key of the device), and the signature of the license $\sigma_{PK_{P_A}}$ (a signature that can be verified with the public key of the authority P_A in the DRM regime A).

The trust between different parties is reflected in the form of an interoperability contract. This contract defines how content can be transcoded and repackaged in the DRM format adopted by each provider and how the rights described in the licenses are translated. For our interoperability architecture, we propose that any parties that agree to interoperate will delegate certain functions to the DIM. For example when providers P_A and P_B agree on a contract, certain information is then given to the DIM. Therefore we assume that the DIM stores $rek_{A \rightarrow B}$ and $rsk_{A \rightarrow B}$ (the re-encryption and re-signature keys). The meaning of these keys as well as the way to use them will be made clear in Section 5. It is essential however, that the information P_A and P_B share

between themselves and between the DIM should not give away any information regarding the secret key of any party. That is that the DIM learns nothing about the secret keys of P_A , P_B , P_{D_A} or P_{D_B} , and that P_A or P_{D_A} (or in general DRM regime A) learns nothing about the secret keys of P_B , P_{D_B} (or in general, about the secret keys used in DRM regime B) and the DIM and viceversa.

Finally, it is important to note that the acceptance of the certificate of the DIM or any other device by a provider, makes sense only when the security guarantees of the importing or intermediate devices are at least the same as the exporting device. This is because the security of the interoperability framework will depend not only on the security of the protocols used, but also on the security of the weakest link, i.e. the client device that can be most easily compromised. In that case, an adversary simply transfers the content to the weakest device and then tries to compromise that device. Therefore, the providers must define clearly their security guarantees and accept only operator certificates that provide this guarantee.

2.2 Domain Interoperability Manager

The Domain Interoperability Manager has a fundamental role in the future of home networks, as it can greatly facilitate the deployment of DRM interoperability services by third parties who have the incentive to do so. As well as managing and keeping track of the devices in the home network, the DIM achieves media portability by translating DRM packaged digital content between the various DRM systems on different devices. We assume the DIM has regular (or at least periodic) online connectivity and can contact providers associated with the various rendering devices in its home network to obtain configuration tools that would allow local content and license translation between different DRM systems. The details of the administration and execution of the movement of content between devices by the DIM is the main contribution of this paper and will be addressed in more detail in the following sections.

Aside from transferring and translating governed content among devices, the DIM can also improve the user experience by providing better understanding of the licenses associated with purchased content. In particular, the user should understand in practical terms what they are paying for when they purchase digital content. To get access to this service, every time users obtain a new hardware or software rendering device, they add it to their home network by registering it with the DIM, so that the DIM has information on all rendering and copying devices belonging to the home Domain. Then, before purchasing digital content, the user can first download the license terms of the content so that the DIM can inform the user which devices can render the content and how this content can be shared among the domain devices based on the usage rights. This service would greatly improve the user experience by preventing the purchase of content that will not preserve the user's fair use expectations. This is a way to address recent calls to ensure that 'it is "crystal clear" to consumers what freedom they have to use the content they are purchasing' [29].

3. ATTACK SCENARIO AND THREAT MODEL

3.1 Traditional Attacks Against DRM Systems

There are in general three types of attacks for any DRM system: (i) attacks to the DRM protocols, (ii) attacks against the secure storage of client devices, and (iii) attacks against the rendering application. The basic objective of these attacks is to obtain the digital content in an unprotected form.

An example of an insecure protocol is an earlier version of Apple's FairPlay in which an emulator *PyMusique* (now *SharpMusique* [23]) could spoof a legitimate iTunes client when downloading music from the iTunes store. The problem was that there was no mutual authentication between the (trusted) iTunes client and the iTunes music store. A similar problem with authentication of the trusted device is exploited by the Hymn project [19], which also provides a tool to emulate a (trusted) iTunes client as if it was installed in a different computer, in order to obtain the key of the user from the iTunes server.

Attacks against client devices usually attempt to dump the content keys or the unencrypted content from the secure storage. An example of these type of attacks are the *FreeMe* and *DRM2WMMV* tools against Microsoft's Windows Media DRM system.

Attacks against the rendering application were exemplified in [4], where the authors replaced part of the rendering application for the Windows Media DRM system so that once the content is decrypted, it can be captured and saved. A similar kind of attack has also been implemented against iTunes [7].

3.2 Threats Against Interoperability Protocols and Devices

In this section, we define three new attacks for interoperable systems: (i) cross-compliance of devices, (ii) splicing of the content with an illegitimate license and (iii) leakage or direct access to the raw content or the content encryption key on the migration path.

One of the most important concerns for interoperability that we have not seen covered in the literature before, is the problem of cross-compliance between different providers and devices, and in particular the problem of coordinating the updates of different devices. Since it is reasonable to assume that attackers will discover vulnerabilities of certain implementations throughout the lifetime of an interoperability contract, providers need to ensure that all devices (including the DIM) are up to date and compliant in a content migration path. Traditionally the content provider for a given device would verify that the device is patched and up to date before sending the content to it. However, in an interoperability framework, the content provider also needs to ensure that the DIM and the importing device are running up to date versions of their corresponding software and firmware. This is further complicated by the fact that the transfer of content between the exporting device and the importing device can happen off-line.

Besides the compromise of devices in which the attacker gets direct access to the digital content, there are other levels of compromise that should be considered. In particular since we are assuming the redistribution of the content to several other devices that can happen in an off-line scenario

(and therefore the license can be obtained from a -possibly compromised- device), of particular importance are the attacks against the license. One such attack is a *splicing* attack, whereby the adversary either modifies an existing license or fabricates a new license.

Another new major threat is the introduction of the interoperability protocol itself. An eavesdropper should not be able to learn any secret information from the interoperability protocol. Furthermore, a compromised party should not be allowed to use the interoperability protocol to obtain any other information from the other participating parties.

3.3 Adversary Model

We assume that the attacker has full control over the network. It has the ability to download any content from provider A to any possible client device that follows DRM_A . Furthermore, it can initiate the transfer of content and license from any client following regime DRM_A to any other participating DRM entity as many times as he or she wants. The attacker is also able to manipulate any transmission and attempt to impersonate any trusted device (i.e. the attacker can perform a man-in-the-middle attack).

Definition 1. An attack against an interoperable system is said to be *successful* if the adversary can obtain access to the content in any way that violates the original license or the license of the interoperability contract.

Definition 2. An interoperable system is *secure* against the above described adversary, if in order to break the security of the system, the adversary has to break either the tamper-resistance of a device in the content migration path, or the underlying cryptographic algorithms.

In other words, the interoperability system is not successfully attacked because of exploiting a vulnerability in the protocol used to transfer the content among the different DRM regimes.

4. ASSUMPTIONS AND BUILDING BLOCKS

4.1 Assumptions

We assume that the provider exporting the content P_A , will have one set of rights for devices following DRM_A and a different set for importing devices. Since the focus of this paper is interoperability, and in particular, the process of transferring digital content from device D_A (following DRM_A) to device D_B (following DRM_B), we will only consider the exporting rights in the remaining of the paper.

We assume that the provider encrypts their raw content using a *content encryption key* CEK. Associated with the content are a set of rights defined by the provider which constrains the actions of the consumer. The rights, CEK, as well as other information such as the content identifier are used to construct the license, which is encrypted and signed by the provider.

At manufacture time, each compliant device is given a public/private key pair with the private key stored in tamper resistant memory, and the public key certified by the manufacturer. Although in practice the manufacturer, the provider and the operator of the device can be different, for notation simplicity we assume in the rest of the paper that they are the same. Therefore the only trusted authority of a device under regime DRM_A is Provider P_A

To better guarantee the enforcement of DRM policies, compliant rendering devices can incorporate a tamper-resistant hardware module (such as the Trusted Platform Module [24]), which can provide:

- (i) **Attestation:** authentication of software to another party.
- (ii) **Isolation:** protecting software from being read or modified by other software or hardware.
- (iii) **Sealing:** allowing software to store data that only that software can later retrieve.
- (iv) **Memory Curtaining:** the rendering application (and encoder in DIM) are embedded in the tamper-resistant hardware to prevent leaking of decrypted content while it is being worked on.

Requiring all devices to have these features is a very strong assumption, yet the fact that a given DRM regime only works with tamper-resistant hardware modules or not will carry an important weight from the point of view of other providers who have to decide to join the interoperability architecture or not. For example, in order to ensure a certain level of security, a given provider who adopts the interoperability architecture, will allow its content to be exported only to devices that contain tamper-resistant hardware.

4.2 Cryptographic Building Blocks

We use some of the following cryptographic building blocks in our proposed approaches. A **proxy re-encryption** scheme [3, 1], allows a semi-trusted proxy to transform a ciphertext computed under Alice's public key, C_{PK_A} , into one that can be decrypted by Bob's secret key, C_{PK_B} , such that the proxy does not obtain any information about the plaintext. In a **proxy re-signature** scheme, a semi-trusted proxy acts as a translator between two parties and converts a perfectly valid and publicly-verifiable signature from Alice on a certain message, $\sigma_A(m)$, into a signature from Bob on the same message, $\sigma_B(m)$. However, the proxy does not learn any signing key and cannot sign arbitrary messages on behalf of either Alice or Bob. Proxy re-signatures were first proposed by Blaze et. al. [3], but more efficient and practical schemes were proposed recently by Ateniese and Hohenberger [2].

Proxy re-signature and re-encryption are useful tools because they allow the DIM (a semi-trusted proxy) to convert a signature or a ciphertext computed under one key (say the public key of device A) to another (say the public key of device B), without the proxy learning any information about the plaintext message or the secret keys of the delegating party. Therefore, for re-encrypting, the DIM does not need to have access to the raw digital content to encrypt the content under a new public key. Proxy re-signatures are also useful because the DIM cannot sign any arbitrary message. The DIM can only re-sign messages that were already signed by a higher authority. Moreover, the devices are guaranteed that the only way the DIM could have forged a signature or decrypted a ciphertext is by breaking the underlying signature or encryption scheme.

An advantage of using proxy re-signatures is that license signatures can be constructed such that the signatures are verified using the associated operator's public key. For example device D_B can verify a translated signature using only the public key of its content provider (provider P_B), and as described in the trust model section, provider P_B is the only authority for device D_B , thus, the DRM of the importing devices need not be modified.

M	Plaintext of protected content
CEK	Content encryption key
R	Rights associated with content M
$\{M\}_k$	If k is a public (secret) key, $\{M\}_k$ is the encryption (signature) of message M using key k
$KeyGen_X(\cdot)$	CEK Key generator algorithm for Device X
$SEnc_k(\cdot)$	Symmetric key encryption algorithm using key k
$Sign_k(\cdot)$	Signing algorithm using key k
$Re-Enc_k(\cdot)$	Proxy re-encryption algorithm using key k
$rek_{A \rightarrow B}$	Re-encryption key used by a proxy to convert an encryption under PK of A to one under PK of B .
$PEnc_k(\cdot)$	public key encryption algorithm using key k
$Re-Sign_k(\cdot)$	Proxy re-signature algorithm using key k
$rsk_{A \rightarrow B}$	Re-signature key used by a proxy to convert a signature under PK of A to one under PK of B .
PK_i	Public key of principal i
SK_i	Secret key of principal i
P_i	Provider i
D_i	Device i
$cert_i$	Certificate signed by the public key of party i

Figure 2: Our Notation

5. PROPOSED PROTOCOLS

We present detailed implementations of two protocols that allow providers to commit to different levels of trust in the DIM, while limiting the modifications required to the exporting and importing DRM systems and devices. The first protocol, minimizes the provider’s trust in the DIM by not allowing it access to the unprotected content. The protocol translates the DRM packaged content, by assuming that the exporting and importing DRM systems support similar encryption and signature schemes as well as identical content and licensing formats. These assumptions can be weakened and completely removed as shown in the latter protocol, by potentially allowing the DIM access to the raw content. Figure 2 defines the notation we use to describe the proposed protocols.

Interoperability protocols in general, consist of three phases. In the first phase, *device discovery*, the various devices in the home network mutually authenticate with the DIM and create secure channels via which they can communicate securely. This can be achieved using various certificate based device authentication protocols such as SSL or [13]. The DIM then registers these devices with their given domain and obtains necessary certifications from the providers.

In the second phase, *device compliancy* is checked by executing a remote attestation protocol between an attesting party and a verifying party. This phases secures the overall protocol against any cross-compliancy attack. The hardware of the device of the attesting party generates a certificate stating the software version that the device is executing. The verifying party can thus authenticate the validity and correctness of the software of the attesting party.

The third phase (and the crux) of the interoperability protocol, specifies how the DRM protected content is translated from an upstream DRM device so that it is compatible to a downstream DRM device, via the DIM. We focus on this step in the following section.

5.1 Protocol 1

The first protocol minimizes the provider’s trust in the

DIM during content and license translation by disallowing it access to the unprotected content (unless the DIM can break the underlying cryptography). This can be achieved only if we assume that the upstream D_A and downstream D_B devices render similar content format and that the exporting and importing DRM systems use similar encryption and signature algorithms and identical rights expression language. To translate the protected content, the DIM only needs to translate the license so that it is encrypted under the public key of the downstream device and signed using the public key of the downstream provider.

Consider a proxy re-signature scheme which translates a signature from provider P_A on a certain message m into one from provider P_B on the same m . Let $rsk_{A \rightarrow B}$ be the re-sign key that is stored on the DIM and let $Re-Sign(\cdot)$ be the re-signature function. Also, consider a proxy re-encryption scheme which translates a ciphertext encrypted under the public key of Device A , on a certain message m into a ciphertext of m encrypted under the public key of Device B . Let $rek_{A \rightarrow B}$ be the re-encryption key that is stored at the DIM and let $Re-Enc(\cdot)$ be the re-encryption function. Finally, and for all subsequent protocols, assume M is the protected content with identifier ID_M and R are a set of rights with identifier ID_R . The use and practicality of identifiers will be discussed in more detail later.

As shown in Figure 3, the upstream device D_A obtains the encrypted content $\{M\}_{CEK}$ and the associated encrypted and signed license $\{CEK, R\}_{PK_{D_A}}, \sigma_{PK_{P_A}} = \text{Sign}_{SK_{P_A}}(CEK, R)$ from its provider. The DIM is used to modify this content so that it can be rendered on a downstream device D_B . The DIM uses the appropriate proxy re-signature $rsk_{A \rightarrow B}$ and proxy re-encryption $rek_{A \rightarrow B}$ keys to convert the encryption key and signature of the license R without revealing it.

5.1.1 Security Analysis

The proxy re-encryption and proxy re-signature algorithms allow us to treat the DIM as a semi-trusted device. If an adversary compromises the DIM, the best it can do is obtain the re-signature or re-encryption keys (specific to that DIM), which do not yield any information on the private keys of either provider A or B , nor the devices D_A or D_B . The proxy algorithms also do not yield any information on the unprotected data in the process of translating the encryptions and the signatures in the migration path.

By encrypting the data, we ensure content and license privacy, while the signature σ guarantees the integrity of the content and the license and binds them together. This secures the protocol against splicing.

Therefore unless the adversary breaks the underlying cryptographic blocks, a DIM-compromising adversary cannot successfully attack the system.

5.1.2 Further Considerations

In this section, we consider some of the practical considerations of Protocol 1. The re-encryption key depends on the keying information of individual devices. In order to make the system more scalable, each device can calculate their own corresponding re-encryption key. This can be done by using a public key and their own key information, which they can then delegate to the DIM (after undergoing the authentication protocol).

The DIM needs to contact provider P_B to obtain its re-signature key. This can be done during the device discovery

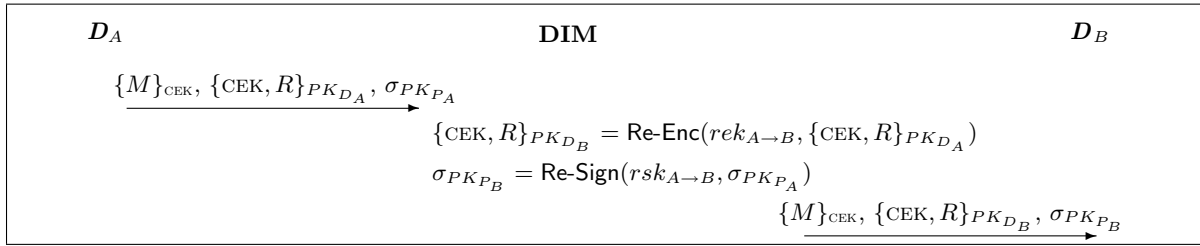


Figure 3: Protocol 1 - the DIM translates the encryption and signature keys of the license from the downstream provider and device to the upstream provider and device respectively. The upstream provider signs the license $\sigma_{PK_{P_A}} = \text{Sign}_{SK_{P_A}}(\text{cek}, R)$ for the upstream device, under its own public key.

phase, when the DIM aggregates information about the various devices in its home network and contacts each provider to obtain DRM configuration details as well as re-signature keys. This is simply a one-time event and therefore the disadvantages associated with connected-oriented interoperability, such as scalability and the assumption of connectivity, do not apply here.

A consequence of using the proxy re-encryption and re-signature schemes is that device D_B can process the DRM protected content and license as though they had been received from provider P_B . This implies no change on the side of the importing device which is a great advantage. Similarly, the software of the exporting device need not be modified to account for interoperability.

The protocol is attractive to a provider because it does not require them to place a lot of trust on the intermediate DIM: even if a DIM is compromised, it cannot reveal the protected content. However, the protocol requires strong assumptions about the exporting and importing devices and DRMs. Firstly, it assumes that the upstream and downstream devices render similar content format. This is reasonable as for instance, most if not all portable music players play the mp3 formats. Protocol 1 also assumes that the exporting and importing DRM systems use identical symmetric and asymmetric encryption and signature algorithms. Most DRM systems, such as Fairplay [20] and Windows Media DRM [26], use the AES encryption algorithm to encrypt their content and RSA based signature and asymmetric encryption schemes to sign and encrypt their licenses. The assumption of identical rights expression languages however is rather strong due primarily, to the different business models providers adopt, and therefore in the following section, we extend our approach to a more flexible setting.

5.2 Protocol 2

Protocol 2 is motivated to allow providers to offer interoperability while maintaining flexibility over their own DRM systems. Specifically, the protocol supports diversity in content format, REL and encryption algorithms. The protocol does assume that the upstream and downstream DRM systems support the same signature scheme. This assumption is necessary to be able to guarantee security of protocol against splicing as the signature bounds the content to its rights and we want this to remain unchanged in the content migration path.

Let k be the session key established between Device D_A and the DIM. Assume that there exists a proxy re-signature scheme as defined in Section 5.1 and also the DIM has pre-

registered with the downstream provider and obtained a certificate cert that certifies its public key PK_{DIM} . To ensure that the protocol is secure against the splicing attack, we assume that the rights and the content have a rights identifier ID_R and a content identifier ID_M respectively. We discuss generating and managing identifiers in more detail below.

As shown in Figure 4, the exporting device D_A obtains from its provider P_A , the encrypted content, the encrypted rights object $\{\text{CEK}_A, R\}_{PK_{D_A}}$ and its signature, $\sigma_{PK_{P_A}} = \text{Sign}_{PK_{P_A}}(\text{CEK}_A, R)$, as well as the identifiers' signature $\delta_{PK_{P_A}} = \text{Sign}_{PK_{P_A}}(ID_M, ID_R)$, where ID_M, ID_R are the content and rights object identifiers respectively. To allow the DIM to transcode the content while not revealing its own private key, D_A encrypts the rights object with the session key k it establishes with the DIM. On receiving the encrypted content and rights object as well as the two signatures, the DIM decrypts the rights object to obtain the content encryption key CEK_A which allows it to decrypt the content M (if transcoding is necessary). The DIM can now transcode the content to the format acceptable to the downstream device D_B . Similarly the rights can be translated to an REL supported by the downstream DRM. CEK_B is randomly chosen from the key space of the encryption scheme of the downstream DRM and the transcoded content is encrypted under this key. The rights are encrypted using the public key of D_B and the license is signed under the public key of the DIM, which is certified by P_B . The DIM uses its proxy re-signature key to convert the signature δ of the content and rights identifiers. Finally the DIM sends to the importing device the encrypted content and license as well as the signatures.

The protocol presented is flexible and can be modified depending on how different the upstream and downstream DRM systems are. In the following we show the modifications that can be made if the upstream and downstream DRM devices have the same:

- (i) **Content Format:** the DIM does not need to access the unprotected content M , therefore $\{M\}_{\text{CEK}_A}$ is not decrypted.
- (ii) **Symmetric Encryption Scheme:** the DIM does not need to select a new CEK and can encrypt M' under CEK_A . However because the translated rights might be different, we cannot use proxy re-encryption to hide the content from the DIM.
- (iii) **Asymmetric Encryption Scheme:** If (i), (ii) and (iv) hold, then we can replace the encryption of the license with proxy re-encryption and prevent the DIM to access the unprotected content completely. This is equivalent to Protocol 1. Otherwise, the protocol cannot be further simplified.

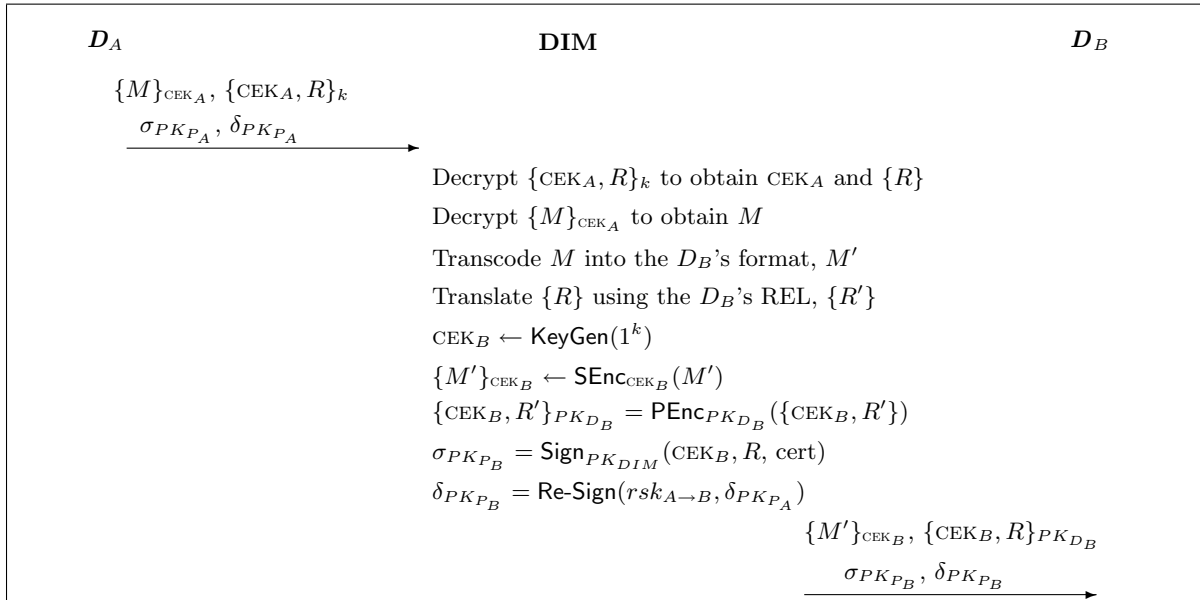


Figure 4: Protocol 2 - An interoperability protocol for two heterogeneous DRM systems. D_A obtains from its provider P_A , the encrypted content, the rights object $\{\text{cek}_A, R\}_{PK_{D_A}}$ and their signature, $\sigma_{PK_{P_A}} = \text{Sign}_{PK_{P_A}}(\text{cek}_A, R)$, as well as a signature $\delta_{PK_{P_A}} = \text{Sign}_{PK_{P_A}}(ID_M, ID_R)$, where ID_M, ID_R are the content and rights object identifiers respectively.

(iv) **REL:** If (i) holds, the DIM can use the proxy re-signature scheme to convert the license signature so that it can be verified under the public key of P_B and avoid using its own key and a certificate chain. The advantage of this is that the importing device software need not be changed at all. If (ii) holds as well, the DIM does not need to get access to the license.

5.2.1 Security Analysis

In this protocol, the DIM does not get access to the private key of neither devices nor providers. The proxy re-signature key is independent of the upstream and downstream providers by definition. Therefore an adversary who gets access to the key cannot gain any information on the system secrets.

The DIM ensures that the proxy re-signature or proxy re-encryption key that it uses is valid by an initial registration phase between the DIM with the various providers and devices in the system. Therefore an adversary cannot simply generate its own proxy re-encryption key for example, to translate the encryption to one under its own key.

In Protocol 2, the DIM has access to unprotected content, either directly when decrypting the content to be transcoded, or indirectly when the license is decrypted so that the rights can be translated. Therefore an importing provider must ensure that the DIM has a trusted computing platform so that it can translate the content in a trusted zone and stores the decrypted content encryption key in safe memory. This will ensure that the content is not leaked during its migration path.

Protocol 2 is also secure against the splicing attack because the content and the license are bound by means of δ which is a signature over the content and the rights identifiers. This signature is initially generated by P_A under its own public key. The DIM, acting as a proxy for P_B , can

translate the signature using the re-signature-key $rsk_{A \rightarrow B}$. As a consequence, D_B can verify that the content and the rights identifiers are correctly coupled by verifying the signature $\delta_{PK_{P_B}}$.

5.2.2 Further Considerations

The advantage of using the proxy re-signature scheme is that it allows D_B to obtain a license signed under the public key of its own provider, P_B . Therefore the importing device does not need to modify its application or its trust relationships. However, if the license is signed under the public key of the DIM, the software on the importing device must be modified to allow the verification of DIM's certificate cert .

An important consideration in this scheme is the practicality of providing identifiers not only for content but also for rights. We do this because we want to secure the scheme against splicing attacks where a malicious DIM tries to pass off an incorrect pairs of content and license. Associating contents with a unique identifier has been considered before by [13, 9]. This can be done in a variety of ways such as hashing the content, including meta data in the content file, inserting the identifier into the content via watermarking or having a global mapping of identifiers to various content.

Generating and managing rights identifiers is more complicated when the importing and the exporting DRM regimes have distinct RELs. In particular, when a set of rights expressed in REL_1 do not have an equivalence in REL_2 , it is difficult to associate the same rights identifier with both sets of rights. Therefore, further work must be done to come up with a systematic method in assigning identifiers to rights. For example, providers must come up with a global mapping of rights/identifiers across all rights expression languages.

It is also important to point out that the translation of licenses is a difficult job because the set of permissions and constraints available in one rights expression language may

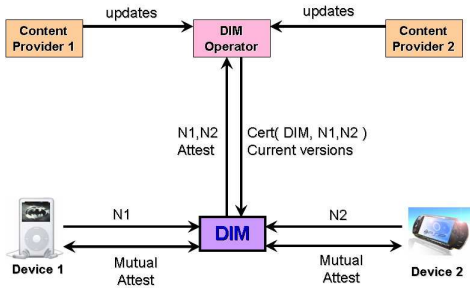


Figure 5: Online Attestation Protocol

not the same as those available in another, and even when equivalence exists, finding a mapping between the two languages might be difficult. Translation of contracts between different providers is out of the scope of this paper, and we refer the reader to [14] for more details.

6. REMOTE ATTESTATION

As we mentioned in our security considerations, it is very important for the exporting device (and the DIM) to check the compliancy of the importing device, ensuring cross-compliancy in the migration path. Note that this problem is different from the revocation problem, since we assume that the attacker does not have access to the secret key of D_B , but is still able (by exploiting some vulnerability) to obtain unprotected digital content from it. In order to correct these vulnerabilities, we assume that the operators of each device issue firmware or software updates to patch the devices. In particular, we assume that whenever D_A tries to obtain new content from provider A , D_A is required to be running the latest version of the software. The problem for our framework is that an attacker can keep the vulnerable device unpatched in order to use it as a tool for extracting protected digital content (by importing the digital content from the exporting devices to the unpatched device). It is important therefore, that each device is assured that the party it is communicating with is running its latest software version. A way to achieve this, is through remote attestation. Remote attestation works by having the hardware of the device generate a certificate stating the software version that is currently running. The verifying party must however know the correct software version that should be running on the device it is attesting.

We propose two approaches of how the devices can obtain the information regarding these software updates. The two approaches differ on how this information is sent to each of the devices. The first approach (offline) uses periodical online updates from the DIM, but it is not required to go online for every transfer across DRM regimes. In contrast, the second approach (online) requires an online connection for every transfer of digital content among devices.

6.1 Offline Attestation

Whenever a new version of the software is released by either device operator (P_A or P_B), a certified notification of this event is sent to the DIM operator. Whenever the DIM operator updates the software of the DIM device, a certified notification is sent to operators P_A and P_B . (i) The DIM

device contacts the online DIM operator periodically (given a predefined period) and downloads the information on the latest software for each of the devices of the domain.

(ii) Every time new protected content is acquired by D_A , provider P_A sends the latest version of the DIM device along with the DRM packaged content and license.

(iii) When content wants to be transferred from D_A to D_B , DIM and D_A attest each other based on their latest information. Similarly, DIM needs to attest D_B using its latest attestation information.

While this approach tries to minimize the number of online connections required, it does however, have some limitations. Firstly, it requires a secure clock in the DIM device to ensure it contacts the DIM operator at the appropriate at all. Secondly, there will always be a delay based on the intervals at which the DIM device contacts the DIM operator. During this lapse of time, the attacker can potentially transfer new content to a vulnerable device. Finally, this protocol places an extra burden on P_A because it needs to provide information regarding DIM devices to its clients.

In order to solve these problems, we now propose a second attestation protocol that transfers this added burden to the DIM. The protocol requires the DIM to have online connectivity for every translation and transfer across different DRM regimes.

6.2 Online Attestation

Figure 5 shows the proposed online attestation protocol. Similar to the the previous approach, assume that whenever a new version of the software is released by either device operator (P_A or P_B), a notification of this event is sent to the DIM operator. Now assume that when the DIM device receives a request to transfer content from D_A to D_B it also receives nonces N_A and N_B (from D_A and D_B respectively). Then once the DIM device and the DIM operator have authenticated each other and established a secure channel, they perform the following steps:

(i) DIM Operator checks compliance of DIM device. If the DIM needs to be updated, it send a request for updates to the user and then aborts. Else it sends N_A, N_B to the DIM operator.

(ii) DIM operator then append N_A, N_B to the attestation of the DIM and signs this message with its secret key to obtain $\sigma_{attest} = \text{Sign}_{SK_{ODIM}}(att, N_A, N_B)$. Finally, it sends to the DIM ($att, N_A, N_B, \sigma_{attest}$) as well as the current software versions from P_A and P_B (with their corresponding certificates).

(iii) DIM performs an attestation of D_A and D_B based on the information received from the DIM operator. If either of them fails attestation, request to the user an update and abort. Else send the attestation report att and σ_{attest} to both devices: D_A and D_B .

Once each of the domain devices has authenticated each other's identities and software versions, we need to start the transfer of the digital content.

7. CONCLUSIONS AND FUTURE WORK

We believe this paper provides the first steps towards a more rigorous analysis of interoperability among different DRM regimes in three key areas. The first area is the analysis of the security reduction of a DRM regime by joining a

new interoperability coalition. The second area deals with incentives for a DRM regime to accept a given interoperability specification, which in our case is approached by minimizing the number of changes required by a DRM regime in order to join the interoperability coalition. The final area we wanted to address is the need for a more detailed presentation of the interoperability protocols in academic papers. This will not only allow researchers to study more rigorously any interoperability proposal, but will help educate future DRM researchers.

An essential part of our protocols is the notion of the DIM: a device acting as a semi-trusted off-line ² translator across DRM regimes. Since the DIM should be able to generate valid licenses for the DRM regime of an importing device, we designed our protocols with the intention of allowing the DIM only to create valid licenses. With our proposed framework content providers can still use their own DRM systems with their devices and achieve a higher market value by allowing interoperability. We note that some of our assumptions may be restrictive for existing DRM implementations, such as using signature schemes for which proxy re-signatures exist. However, the aim of the presented architecture is to show how off-line interoperability can be achieved with high security guarantees as long as some standard components (such as the selected signature schemes) are satisfied. The promise of interoperability and its associated requirements will then be another factor to consider by providers in their selection of cryptographic algorithms and REL's to use in their DRM regime.

We also presented security requirements and threat models. However, DRM systems are very complex entities due to all the details that need to be considered (cryptographic protocols, business models, rights expression languages, trusted computing etc.), and therefore a formal security analysis cannot be easily achieved. We believe however that our security and threat models are promising first steps towards a more rigorous security analysis.

One of the problems we did not address in this paper, is the problem of revocation. In revocation we need to assume that the adversary can fully compromise (obtain the secret encryption keys of any device) and still pass the attestation process. We plan to look at the revocation problem in the future. We are also planning to compare our framework, which was based on a public key infrastructure, to other possible techniques used in DRM systems, such as broadcast encryption, and analyze the efficiency and security tradeoffs that such a technique can offer.

Acknowledgments

This work was supported in part with funds from the Department of Defense as well as the U.S. Army Research Office under Award No. DAAD19-01-1-0494, and the U.S. Army Research Laboratory under Cooperative Agreement DAAD19-01-2-0011 for the Collaborative Technology Alliance for Communications and Networks.

8. REFERENCES

- [1] G.Ateniese, K.Fu, M.Green, S.Hohenberger, *Improved Proxy Re-Encryption Schemes with Applications to Secure Distributed Storage*, NDSS, 2005.
- [2] G.Ateniese, S.Hohenberger, *Proxy Re-Signatures: New Definitions, Algorithms, and Applications*, ACM CCS'05.
- [3] M.Blaze, G.Bleumer, M.Strauss, *Divertible Protocols and Atomic Proxy Cryptography*, EUROCRYPT'98.
- [4] T.Hauser and C.Wenz, *DRM Under Attack: Weaknesses in Existing Systems*, Digital Rights Management: Technological, Economic, Legal and Political Aspects, November 2003.
- [5] N.Herberger, *Virgin Media versus iTunes*, http://www.indicare.org/tiki-read_article.php?articleId=150, October 2005.
- [6] P.A.Jamkhedkar, G.L.Heileman, *DRM Interoperability Analysis from the Perspective of a Layered Framework*, Proceedings of the ACM Digital Rights Management workshop DRM'05, 2005.
- [7] H.L.Jonker and S.Mauw, *Core Security Requirements of DRM Systems*, Symposium on Information Theory in the Benelux, June 2004.
- [8] R.H.Koenen, J.Lacy, M.Mackey, S.Mitchell, *The Long March to Interoperable Digital Rights Management*, Proceedings of the IEEE, vol 92(6), June 2004.
- [9] D.W.Kravitz, T.S.Messerges, *Achieving Media Portability through Local Content Translation and End-to-End Rights Management*, Proceedings of the ACM Digital Rights Management workshop DRM'05, 2005.
- [10] M.Mambo, K.Usuda, E.Okamoto, *Proxy Signatures: Delegation of the Power to Sign Messages*, IEICE Trans. Fundamentals, 1996.
- [11] S.Michiels, K.Verslype, W.Joosen, B.De Decker, *Towards a Software Architecture for DRM*, Proceedings of the ACM Digital Rights Management workshop DRM'05, 2005.
- [12] S.K.Nair, B.C.Popescu, C.Gamage, B.Crispo, A.S.Tanenbaum, *Enabling DRM-preserving Digital Content Redistribution*, IEEE Conference on E-Commerce Technology (CEC'05), 2005.
- [13] B.C.Popescu, B.Crispo, A.Tanenbaum, F.Kamperman, *A DRM Security Architecture for Home Networks*, Proceedings of the ACM Digital Rights Management workshop DRM'04, 2004.
- [14] R.Safavi-Naini, N.P.Sheppard, T.Uehara, *Import/Export in Digital Rights Management*, Proceedings of the ACM Digital Rights Management workshop DRM'04, 2004.
- [15] A.U.Schmidt, O.Tafreschi, R.Wolf, *Interoperability Challenges for DRM Systems*, IFIP/GI Workshop on Virtual Goods, Ilmenau (Germany), May 2004.
- [16] Z.Tan, Z.Liu, *Provably Secure Delegation-by-Certification Proxy Signature Schemes*, ACM International Conference on Information Security, 2004.
- [17] Susan Wegner, *Prototype Description of an Open DRM Architecture*, OPERA-Interoperability of Digital Rights Management Technologies, EURESCOM project report, December 2003.
- [18] DVB - The Digital Video Broadcasting Consortium. <http://www.dvb.org/>
- [19] Hymn Project. <http://hymn-project.org/>
- [20] iTunes FairPlay. <http://www.apple.com/lu/support/itunes/authorization.html>
- [21] Open Mobile Alliance. <http://www.openmobilealliance.org/>
- [22] Secure Digital Container. <http://www.digicont.com/>
- [23] Sharpmusic. <http://www.nanocrew.net/>
- [24] Trusted Computing Group, *Trusted Computing Platform Alliance Main Specification*, February 2002, Version 1.1b, <http://www.trustedcomputinggroup.org>
- [25] Microsoft Windows Media Rights Manager. <http://www.microsoft.com/windows/windowsmedia/howto/articles/drmarchitecture.aspx>
- [26] Windows Media DRM, wikipedia entry. http://en.wikipedia.org/wiki/Windows_Media_DRM
- [27] *Can we learn from Apple's success with iTunes Music services?*. <http://www.dk.cappemini.com/NewsmailSystem/Telecom/Ver1/Documents/ITunes.shtml>.
- [28] Intertrust's Coral and Marlin. <http://www.intertrust.com/main/research/initiatives.html>
- [29] MPs in digital downloads warning. <http://news.bbc.co.uk/2/hi/technology/5041684.stm>
- [30] <http://www.coral-interop.org/>
- [31] Advanced access content system. <http://www.aacsla.com/home>
- [32] PachyDRM. <http://www.pachydrm.com/>
- [33] The Informed Dialogue about Consumer Acceptability of DRM Solutions in Europe (INDICARE), "Consumer Survey on Digital Music and DRM", May 2005, www.indicare.org/survey

²With an off-line attestation protocol, connection is only sporadic.