A Four-Layer Model for Security of Digital Rights Management

Eric Diehl
THOMSON R&D France
1 Avenue Belle Fontaine
CESSON SEVIGNE, FRANCE
+33 2 99 27 32 54

eric.diehl@thomson.net

ABSTRACT

Defining Digital Rights Management (DRM) is a complex task. There is no unique universal definition. There are many legal, economic, functional, and technical definitions. This complexity induces also that there is not one unique modeling of DRM. Each model should help to compare different DRM systems and easily highlight the differences and the similarities between them. One of the weaknesses of the current models is that none puts specifically the focus on the most important characteristics of DRM: protection of content and rights management. We propose a four-layer model that complements traditional ones. Using trust layer, rights management layer, rights enforcement layer, and content protection layer, this model is security oriented. It is suitable to describe any content protections such as DRM, conditional access, copy protection or even pre-recorded content protection systems.

Categories and Subject Descriptors

D.2.11 [**Software Engineering**]: Software Architectures – *Domain-specific architectures;* K.5.1 [**Legal Aspects of Computing**]: Hardware/Software Protection – *Licensing, Proprietary rights*

General Terms

Security, Design

Keywords

Digital Rights Management, DRM, rights enforcement, OMA, DVB, DTCP

1. INTRODUCTION

Interoperability of Digital Rights Managements (DRM) is currently one of the hottest topics, both in the industrial and academic worlds. Although the requirements are rather well understood [1], no convincing solutions appear. One possible reason is that the security models of different systems are difficult to analyze together. Currently, none of the high-level models of DRM [2] which take into account security issues seem to fit to a large set of heterogeneous DRM and copy protection systems.

Defining DRM is a complex task. There is no unique universal definition. There are many legal, economic [3], functional [4], and technical [5] definitions. This complexity induces also that there is not one unique modeling of DRM. Each model should help to compare different DRM systems and easily highlight the differences and the similarities between them. One of the weaknesses of the current models is that none puts the focus on the most important characteristics of DRM: protection of content and rights management. We propose a four-layer model that complements traditional ones. This model is security oriented. It is suitable to describe any content protections such as DRM, conditional access, copy protection or even pre-recorded content protection systems. For several years, we have been using it for the design, analysis and specifications of many content protection systems.

Section 2 provides a short introduction to the existing models of DRM. Section 3 describes the four-layer model. In section 4, using the four-layer model, we describe three totally different systems. In the end, section 4.4 introduces some potential tracks for future work.

2. RELATED WORKS

To fully model a system, the designer analyses different point of views [6]. This is especially true for complex systems such as DRM. Each model represents one specific point of view. Currently, DRM models are mostly addressing functional, transactional and architectural point of views.

Functional models describe the main functions provided by DRM. It mainly covers three aspects: the management of the rights, the management of the usage and the management of the content [7,8]. Management of rights defines the constraints, the granting, and the commercial conditions attached to content. The management of usage enforces the conditions defined by the provider as the usage rights whereas the management of the content handles the content itself. Thus, functional models define concepts such as usage rights management, content packaging and delivery, monitoring …

The functional model implies modeling the usage rights and their enforcement. This is probably the most studied DRM technology [9][10][11][12]. The goal is to define the most expressive language and proof its completeness. Some studies even attempt to define language supporting the fuzzy notion of fair use [13]. In the field of enterprise DRM, such exhaustive expressiveness may have sense. In the case of commercial DRM, and especially in Business to Consumer (B2C), it is more questionable.

DRM'08, October 27, 2008, Alexandria, Virginia, USA.
Copyright 2008 ACM 978-1-60558-290-0/08/10...$5.00.

Transactional models describe the dynamic behavior for the different steps starting from the packaging of the content and ending to the actual consumption of the protected content [9]. Transactional model puts the focus on the process and its enforcement.

Architectural models describe the different elements of the architecture of a DRM and their interaction [14]. It mainly deals with servers, services and agents. It identifies the technical services to provide and the entities that provide these services. It maps the functional model into the corresponding software and hardware elements. This is the most known type of model. Often descriptions of DRM rely on architectural model.

Presenting a system in a layered model is a common practice in information system. The most known layered model is the 7-layer OSI model for communication. Jamkhedkar and Heileman propose a layered model of DRM compatible with the OSI model [15]. This model is extremely interesting. It uses five layers. The two upper layers fit with their counterparts in the OSI model. This model nicely illustrates the interactions between the servers and the client. They push the analogy with the OSI model extremely far. In their latest work, they even added a physical layer [16] pushing the analogy one step further.

None of these models does purely focus on security aspects of DRM. However, security is one of the main expected characteristics of DRM. Security is the primary focus of the proposed four-layer model.

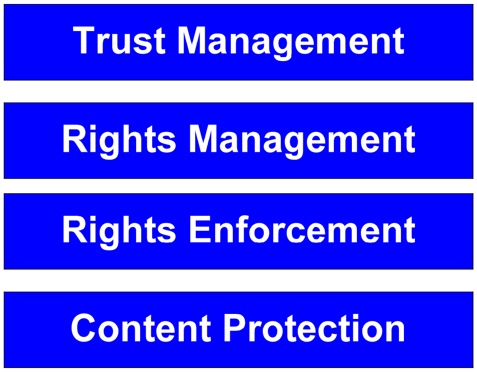

Figure 1: The four layers of the model

3. CONTRIBUTION

Although it presents some similarities with Jamkhedar and Heileman's model, the four-layer model exclusively focuses on security. The two middle layers may use the same vocabulary than within the model of Rump [17]. Nevertheless, the split between management and enforcement is different[1].

The four-layer model describes the behavior of a DRM system through four main security features:

- trust management,
- rights management,
- rights enforcement,
- and content protection.

Figure 2 summarizes the interactions between these four layers. The trust management serves all three bottom layers. Thus, we will explain its contributions after the descriptions of the three other layers (see section 3.4).

3.1 The Rights Management Layer

The rights management layer handles the usage rights associated to one content. According to Rump, this layer manages digital rights [17]. Usage rights express permissions and constraints on one content. Usage rights might be simple, such as Boolean subscription-based access rights, for instance subscription to a service or a bundle of service, or listen once. Modern DRMs use more complex usage rights requiring complex syntactic languages called Rights Expression Languages (REL). They are often based on mark-up languages. The two most known RELs are ODRL [18] and XrML [19]. Other interesting RELs exist such as Script License [13]. All these languages are extremely rich. They can express complex usage rights such as "content can be viewed 4 times during the next day and copied only once but with lower resolution".

On the server side, the rights management layer receives the commercial rights allocated to the content. Content providers or content distributors define these commercial rights from their commercial offer[2]. This layer translates these commercial rights into usage rights using its own syntax. It forwards this information to the rights enforcement layer.

On the client side, the rights management layer receives the usage rights and checks if the consumer is authorized to perform the requested actions. The rights management layer forwards its decision to the rights enforcement layer.

3.2 The Rights Enforcement Layer

The rights enforcement layer ensures that content will be used only under the conditions defined by the usage rights. According to Rump, this layer digitally manages rights [17]. Thus, it has two main roles:

- It protects the usage rights associated with content against tampering. An attacker should not be able to modify the usage rights.
- It guarantees that the usage rights are obeyed and not bypassed. An attacker should not be able to use the content in a way not authorized by the usage rights.

On the server side, the rights enforcement layer encapsulates the usage rights defined by the rights management layer, the secret key used by the content protection layer, and some ancillary data into a data structure. We use the generic term of license to describe the corresponding data structure. It may have different names in commercial applications such as Entitlement Control Messages (ECM), rights object or Digibox™. The license (or at least sensitive parts of the license) is signed and encrypted. Signature prevents alteration of usage rights. Encryption prevents eavesdropping of the secret key protecting the content. The usage rights do not need encryption; they only have to be

[1] In Rump's model, most of these layers would be categorized as enforcement functions of DRM. This model defines more precisely what is enforced.

[2] In case of non commercial DRM, commercial rights are replaced by access rights defined by security policy.

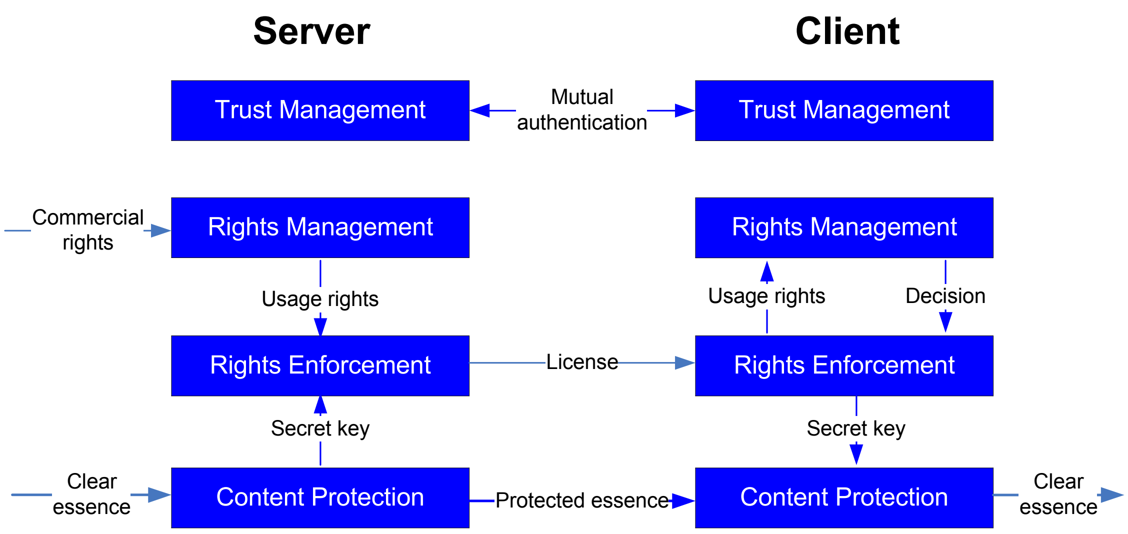

Figure 2: Interaction between layers

protected against alteration. For user interface, it may be useful to have the usage rights in clear text. Stricto-senso, the secret key would only need confidentiality. Nevertheless, if the secret key would be tampered, then the access to content would be denied. It could allow a nice denial of service attack on commercial services.

On the client side, the rights enforcement layer decrypts the license, and checks its integrity with the signature. It forwards the extracted usage rights to the rights management layer that answers with an eventual authorization. If authorized, the rights management layer forwards the secret key to the content protection layer and enforces the constraints defined by the usage rights, for instance requesting the device to reduce the video resolution, or monitoring the duration of listening. The enforcement implies also to protect the integrity of different parameters such as time and date, number of times one content was viewed, identity of the user …

The actual enforcement of the usage rights is a complex task if the usage rights are extremely rich. This task is rarely studied in the academic field. Current models prove that they can formally enforce the expressed rights [12]. Unfortunately, there is no description on how to perform a secure implementation of these formal rules on a non-trusted platform like a PC. As in cryptography [20], despite formal proofs of security, there are many way to break the security of the actual implementations [21].

For instance, how do you protect the states of the usage rights of a given content within the computer memory without tamper resistant hardware? Without online connection? The designer has to prevent replay attack (i.e. restoring the previous memory state), fault injection and interruption when updating the state, reverse engineering software code. This may be one of the weakest points of current computer based DRM system.

Furthermore, modern RELs have a huge expressiveness offering quasi unlimited possibilities. It is obvious that secure implementation of extremely complex usage rights is probably difficult if not unfeasible. If we believe that there is a real need for such complex usage rights, then the scientific community should develop theoretically secure schemes supporting them.

3.3 The Content Protection Layer

The content protection layer is the lowest one in the stack. It securely seals the content so that no attacker could access the content without having the associated rights. This protection is essential because breaking this one circumvents all previous enforcements.

The usual mechanism is encryption. On the server side, the content protection layer scrambles the content using a secret key and a defined encryption algorithm[3]. This algorithm is often referred as bulk encryption. The result is sometimes called secure container. There are mainly two types of scrambling algorithms: for streamed content, encryption uses stream cipher like ISMAcryp, or DVB-CSA, whereas for file based content, it uses block cipher such as AES.

On the client side, the content protection layer can only descramble the content if the rights enforcement layer forwards the right secret key.

The content protection layer may also add limited defense against analog hole [22]. In the digital domain, the only known defense

[3] The use of terms such as scrambling and descrambling to refer to bulk encryption are inheritance from the early Pay TV systems. When content was analog, content was scrambled using different techniques such as variable time delay, line cut and rotate or line shuffling [38]. Often the associated licenses were digitally encrypted. Thus, the usage started to scramble content and encrypt license.

is the enforcement watermark. A watermark carries so called Copy Control Information that a compliant recorder would check before recording.

Another method attempts to mitigate the risk of large scale leakage through analog hole: forensic marking. The content protection layer may alter the content to trace back the consumer. In case of illegal distribution of the marked content, content providers may trace back the source of leakage. The alteration may be visible or invisible. The content protection layer may append to the video or the document a logo or the name of the recipient. The content protection layer may add an invisible watermark carrying the unique ID of the receiver [23]. Forensic marking may occur at the server or at the client side. The first approach has the advantage to be more secure (theoretically, it should be more difficult to tamper the server) but adds strong real time constraints on the server. The second approach is more cost effective. However, it assumes that an attacker cannot tamper the client software, else the attacker may bypass the watermark insertion or modify the data to be marked. Forensic marking is commonly used in post production houses [24].

3.4 The Trust Layer

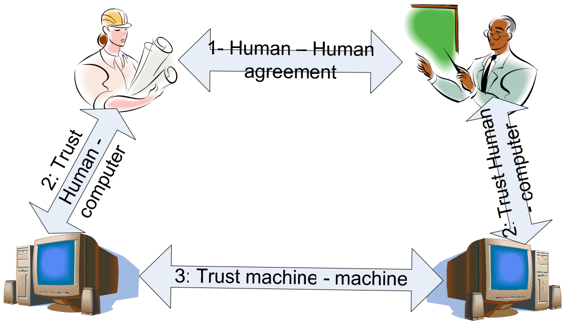
1- Human – Human agreement
2: Trust Human - computer
2: Trust Human - computer
3: Trust machine - machine

Figure 3: Trust relationships

Trust is the cornerstone of any DRM system. Establishing trust relationships is complex [25]. It should turn human – human (or merchant – customer) agreements into machine – machine enforcement as illustrated by Figure 3. This is the role of the trust management layer. The trust management layer ensures that only trustful principals interact. The system should deny any interaction with non-trusted principals, or provide limited services to the non-trusted principals. Trust means two things:

- The principal is what it is supposed to be from the point of view of identity and functionality.
- The principal behaves as expected respecting the compliance rules.

Trust is "materialized" through authentication and certificates[4]. One (or several) authority issues signed certificates for all compliant principals. In addition to cryptographic material, the certificate may also declare capabilities, or requirements of the principals. In other words, via the certificates, the issuing authority acknowledges that the principal owning the certificate is authorized to perform a set of features, and presents a given set of capabilities. The trust layer checks that the certificates are valid, and not revoked.

Only authenticated and compliant issuers should be able to create licenses for trusted clients. Only authenticated and compliant clients should be able to open licenses from trusted servers. Other mechanisms than certificate are available for trust management such as common secret, or common secret function. Then, only the principals that share the same secret trust each other. This common information is provided by the authority.

Each layer verifies trust. The rights management layer may authenticate the identity of the issuers, of the principals, and even that issuers have the rights to generate such usage rights. For instance, the server's rights management layer may check the certificate to know which type of usage rights the client's rights enforcement and management layers may support.

The rights enforcement layer handles the license. It should only handle licenses from trusted entities. In some cases, the content protection layer may share special secret features that partially authenticate each other.

4. EXAMPLES

The four-layer model can describe any DRM technology. More broadly, it is suitable to any system that protects content. To validate this assertion, we will describe three state of the art systems. The first one is a traditional DRM system. We will use a standardized DRM solution: Open Mobile Alliance (OMA). The second type is Pay TV Conditional Access. Pay TV has many common attributes with DRM. Once more, we will use a standardized solution: DVB. The third example is drastically different. It is a link protection system: Digital Transmission Content Protection (DTCP). DTCP does not fit for typical functional, transactional and architecture models of DRM. The objective is to demonstrate that the four-layer model can represent very different content protection systems.

4.1 OMA DRM

OMA was formed in June 2002 by nearly 200 companies including the world's leading mobile operators, device and network suppliers, information technology companies and content and service providers. OMA develops specifications to support the creation of interoperable end-to-end mobile services. [26]. Among these services, OMA defines a DRM to protect Audio Visual content sent to mobile phones. They issued a first version whose target was ring tones protection. With the promise to download movies, or songs, there was a need of a more powerful DRM to protect media contents. This is OMA DRM 2.0 [27].

OMA DRM 2.0 uses a very traditional architecture as illustrated by Figure 4. A content issuer packages and protects media objects into DRM content. It scrambles clear media objects using 128-bit AES with a symmetric Content Encryption Key (CEK). The rights issuer defines the associated usage rights. These usage rights are described using a Rights Expression Language (REL): Open Digital Rights Language (ODRL). ODRL is a XML based language [28]. These usage rights are packaged together with CEK into a rights object. The rights object is cryptographically bound to the DRM content it specifies. Normally, a rights object is associated to one DRM agent. With the notion of domain, a

[4] These certificates are more than typical signed cryptographic certificates.

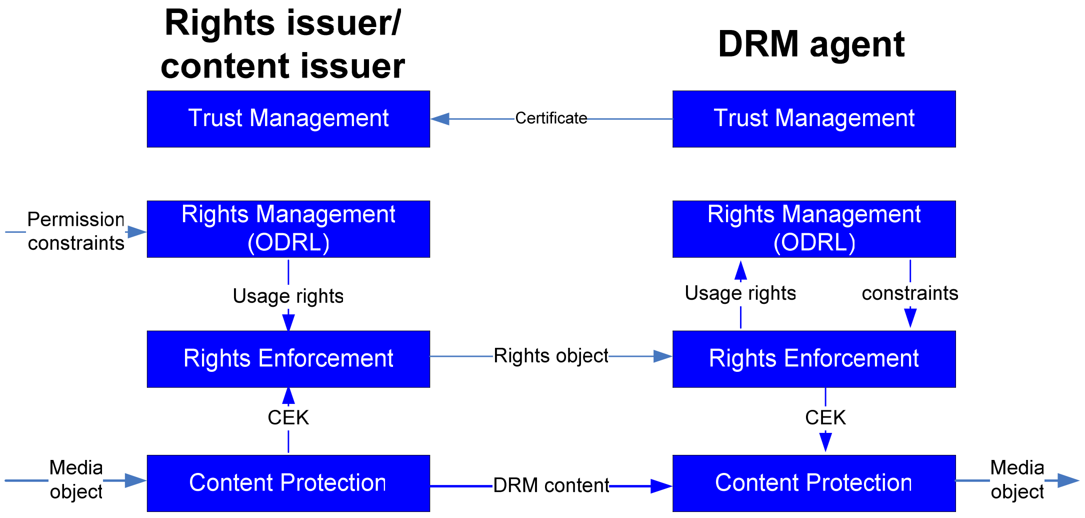

Figure 4: 4-layer model of OMA

rights object may be associated to all the DRM agents belonging to the same domain. Content issuer and rights issuer are roles. Thus, they may be operated by the same entity or by distinct entities.

A DRM agent is the trusted entity that executes on the mobile appliance. The DRM agent receives DRM content and rights object. When trying to play DRM content, the DRM agent opens the associated rights object. This is only possible if the rights object was dedicated to it. The rights object is encrypted so that only the targeted DRM agent can decrypt it. The rights management layer parses the ODRL expression to check the permissions and extracts the constraints. The right enforcement layer ensures the obedience to these constraints and passes CEK to the content protection layer that descrambles DRM content.

Every DRM agent has a unique public/private key pair with a certificate delivered by a Certification Authority (CA). The certificate, in addition to typical Public Key Infrastructure properties, carries also information on the characteristics of the DRM agent. Thus, a rights issuer may decide if it accepts to deliver rights object to a given DRM agent or not. Furthermore, this information allows to encrypt the rights object only for the expected DRML agent. The rights issuer signs every rights object it issues.

Suitability of the four-layer model

Mapping of OMA DRM to four-layer model is extremely straight forward. The four-layer model was designed to fit with DRM. The model shows the importance of the trust management layer. Even if the DRM agent uses ODRL, supports all business models and implements the standardized bulk encryption (AES), it is not sufficient to receive content from any OMA merchants. The merchant and the DRM agent have to share the same trust layer, i.e. the same CA. OMA does not provide one CA. So called compliance regimes will provide their own CA. Compliance regimes define a set of additional rules, so called compliance and robustness rules that a device has to comply with [29]. In other words, a compliance regime defines a given trust layer for a system. Traditional models do not clearly illustrate this dependency. The fact that OMA may have different trust layers, due to different compliance regimes, may hinder OMA's interoperability. This is in contradiction with common belief. If two compliance regimes are not interoperable, then devices will not interoperate.

4.2 Conditional Access System

To illustrate conditional access system, we take the Digital Video Broadcast (DVB) system. DVB is an industry-led consortium of over 270 broadcasters, manufacturers, network operators, software developers, regulatory bodies and others in over 35 countries committed to designing global standards for the global delivery of digital television and data services. Services using DVB standards are available on every continent with more than 120 million DVB receivers deployed [30].

In the early 90s, Pay TV was starting. Many European broadcasters were competing. Several Conditional Access providers were competing. Broadcasters were supporting the cost of Set Top Boxes and of the broadcast equipment. It was rapidly clear that price reduction could only happen if there would be a mass market. Thus, it was mandatory to standardize as many elements as possible. DVB is consensus driven. Thus, an adopted solution is a finely crafted equilibrium that respects at maximum the interests of all stakeholders. Thus, DVB standardized the format of the video selecting MPEG2. DVB standardized the way to signal on the air programs (DVB SI).

DVB defined the scrambling algorithm used to protect the content. It was the DVB Common Scrambling Algorithm (DVB-CSA). DVB-CSA uses a 40-bit key called Control Word[5]. The

[5] New version called CSA2 uses 56 bit. Due to the short life time of the key, it is assumed to be long enough.

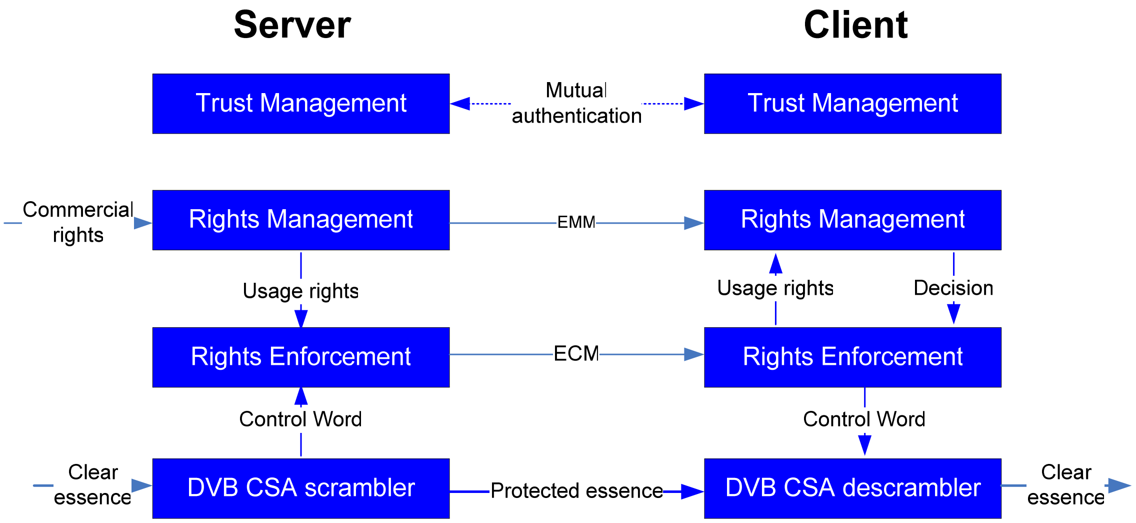

Figure 5: 4-layer model of DVB scheme

Control Word is valid for a given period, called crypto period. DVB-CSA's crypto period can vary from 10 to 120 seconds. A hardware device scrambles the clear video stream. A dedicated smart card, often called the mother card or master card, generates dedicated messages so called Entitlement Control Messages (ECM). ECM contains at least the Control Word of the crypto-period[6], the identification of the program, and the usage rights needed to access the program. The Subscriber Management System handles the commercial rights of the customers. It generates dedicated messages so called Entitlement Management Messages (EMM). EMM contains the identification of the targeted customer or group of customers and the new allowed usage rights. ECM and EMM are encrypted using secrets dedicated to a given broadcaster (or group of broadcasters). The scrambled video carries ECM and EMM. Sometimes, it is possible to send EMM through other delivery channels such as phone lines.

The set top box has an associated smart card. It receives the scrambled video, the ECM and EMMs. The set top box forwards the encrypted ECM and EMM to the smart card. If the smart card belongs to the right broadcaster and is not revoked, then it has the secrets needed to decrypt the ECM and EMM. It means that the trust management is performed if both the head end and the smart cards share some secret information. When the smart card receives an ECM, it decrypts it and then checks if viewing is authorized. For that purpose, it checks the requested rights, carried by ECM, with the ones stored in its protected memory. If the rights are granted, then the smart card returns the control word to the set top box. The set top box can then descramble the video stream. When the smart card receives an EMM, it decrypts it and then checks if it is one of the expected addressee. If it is the case, then it updates correspondingly the stored usage rights.

The smart card hosts the rights enforcement layer, the rights management layer and the trust layer. This architecture has the advantage that it is easy to renew these layers by simply updating or changing the smart card. In fact, DVB has specified only the content protection layer, the identification of ECM and EMM, their size and the way to carry them in the stream. The actual behavior, specifications and security are fully proprietary for each conditional access provider.

Suitability of the four-layer model

The four-layer model is a useful tool to describe conditional access system. Furthermore, this model highlights the strong similarities that exist between DRM and Conditional Access Systems. Other models do not highlight it. The main difference is that the license is split into two licenses. The ECM is equivalent to the DRM license. The EMM updates the rights stored on the client side. This is the sine qua none condition to work off-line (or at least without a back channel to license server).

The four-layer model illustrates the interoperability that DVB introduced with Simulcrypt and Multicrypt. All DVB set top boxes share the same content protection layer and the same interface to the rights enforcement layer. The smart card hosts the three upper layers. This approach allows several conditional access providers to share the same set top boxes although they use proprietary secret systems.

[6] In fact, ECM contains in most cases both the control word of the current crypto-period and the control word of the next crypto-period. This allows smooth transition between crypto-periods.

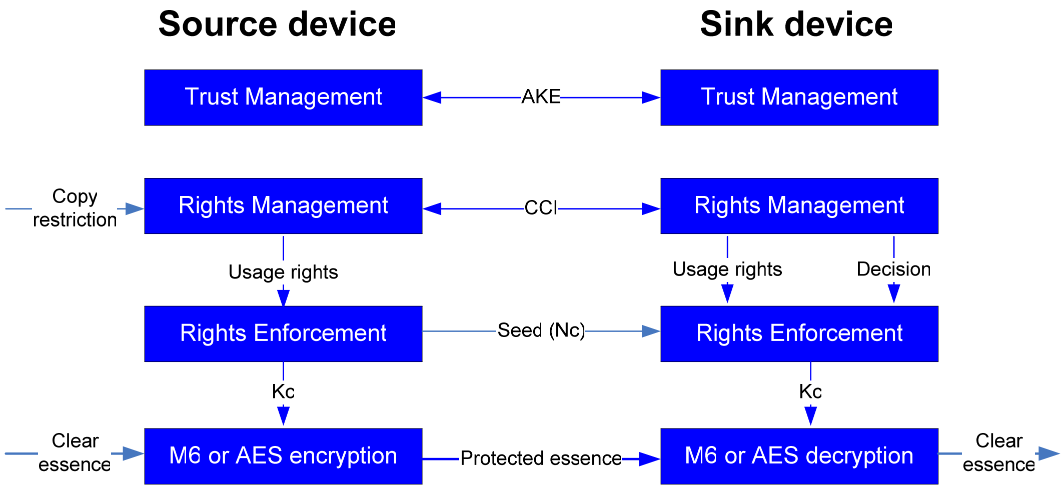

Figure 6: 4-layer model of DTCP

4.3 Digital Transmission Content Protection

Hitachi, Intel, Matsushita, Sony and Toshiba designed the first version of DTCP in 1998. DTCP protects audio video content while it traverses digital transmission mechanisms such as IEEE 1394, or Ethernet IP [31]. Furthermore, DTCP handles basic copy usage rights.

When a source device sends content to a sink device, the first step is mutual authentication. DTCP supports two types of authentication. The full authentication is for devices that support all the copy restriction modes. It uses Elliptic Curve Digital Signature Algorithm (EC-DSA) and Elliptic Curve Diffie Hellman (EC-DH) in a typical challenge response protocol. The restricted authentication is for devices with fewer capabilities. It uses a shared secrets scheme. Both authentication schemes define a common exchange key.

Once the source and sink devices mutually authenticated, the source sends scrambled content to the sink. For that purpose, it may use either M6 [32] or 128-bit AES algorithms. The content key K_c used to scramble/descramble is defined by the source device. It is the result of a function using the common exchange key, a random seed generated by the source, and a value depending on the value of the Copy Control Information (CCI). The CCI has four possible states: Copy Never, Copy One Generation, No More Copies (that is the result of copy of a Copy One Generation instance), and Copy Freely. The CCI is embedded in the content stream. The source device sends the seed to the sink device. The rights enforcement layer at sink device combines the received seed, the received CCI and the exchange key to recover the content key K_c. If no data were impaired, then the sink can calculate the right value of key K_c and decrypt the content. CCI information is forwarded to the device to decide what to do with the descrambled content. For instance, a recorder will refuse to record Copy Never and No More Copies content.

<u>Suitability of the four-layer model</u>

It is impossible to map DTCP to traditional DRM models. DTCP shares only a reduced set of goals with DRM: Protecting the illegal duplication of content during digital transfer. Nevertheless, the four-layer model can describe the behavior of DTCP.

Furthermore, the four-layer model is useful to analyze the interaction of a DRM that imports the content in a device and DTCP that carries this content to other devices on the home network. It is obvious that if a DRM exports content to DTCP then two layers will have to interoperate:

- DRM rights management layer will have to translate its rich usage rights into the four potential usage rights supported by DTCP rights management layer
- The two content protection layers will have to exchange. We may assume that the bulk encryption differs. In that case, either they exchange clear content, or the DRM content protection layer will have to transcript from its bulk encryption into M6.
- If in the same device, the interaction between the trust layers may be null due to intrinsic trust.

The four-layer model allows defining the way two heterogeneous content protection systems may interact. Furthermore, it allows highlighting the parts that may require special care due to higher risk.

4.4 Work to Continue

One of the hottest current topics in DRM field is interoperability [1]. The four-layer model is an interesting tool to study the possible interoperability between DRM systems. For instance, the four-layer model can easily explain the already existing interoperability between DVB decoders using DVB Common Interfaces [33]. All DVB set top boxes share the same content protection layer and interface to the rights enforcement and management layer.

Similarly, the four-layer model highlights that defining a common set of usage rights and a common Rights Expression Language is not a sufficient condition to create interoperability. The other layers have also to interoperate by some means.

We are currently studying the current interoperability initiatives such as OMA, DVB-CPCM [34], CORAL [35], DMP [36], or DReAM [37]. We will extend the work of Heileman and Jamkhedkar [16] by using the four-layer model. We expect to find a new taxonomy of interoperability (based on the four-layer model) and perhaps even find new innovative approach.

5. CONCLUSIONS

We present a new model for DRM and content protection scheme. The four-layer model identifies four main security problems in DRM: the management of the trust, the management of the digital rights, the enforcement of these digital rights, and the protection of the digital content. This model complements current functional, transactional, and component base models.

This model is useful in the specification phase of a content protection system. It is also useful to study how two heterogeneous content protection systems can interact. Thus, this model may be useful to study interoperability of DRM.

6. REFERENCES

[1] R.H. Koenen et al., “The Long March to Interoperable Digital Rights Management,” *Proceedings of the IEEE*, vol. 92, 2004, pp. 883-897.

[2] W. Rosenblatt, S. Mooney, and W. Trippe, *Digital Rights Management: Business and Technology*, John Wiley \&amp; Sons, Inc., 2001.

[3] A.G. GEFFROY, “Economic analysis of Copyright laws and DRMs,” Feb. 2006; http://www.ist-ipmedianet.org/Economic_Analysis.pdf.

[4] “DIGITAL RIGHTS Background, Systems, Assessment,” Feb. 2002; http://ec.europa.eu/information_society/eeurope/2005/all_about/digital_rights_man/doc/workshop2002/drm_workingdoc.pdf.

[5] “Digital Rights Management,” *Wikipedia*; http://en.wikipedia.org/wiki/Digital_rights_management.

[6] G. Booch, J. Rumbaugh, and I. Jacobson, *The Unified Modeling Language User Guide*, Addison-Wesley Professional, 1998.

[7] B. Furht, E. Muharemagic, and D. Socek, *Multimedia Encryption And Watermarking*, Springer, 2005.

[8] R. Iannella, “Digital Rights Management (DRM) Architectures,” *D-Lib magazine*, vol. 7, Jun. 2001; http://www.dlib.org/dlib/june01/iannella/06iannella.html.

[9] S. Guth, *Interoperability of Drm Systems: Via the Exchange of Xml-based Rights Expressions*, Peter Lang Pub Inc, 2006.

[10] C.N. Chong, S. Etalle, and P.H. Hartel, *Comparing Logic-based and XML-based Rights Expression Languages*, Springer, .

[11] C.A. Gunter, S.T. Weeks, and A.K. Wright, “Models and languages for digital rights,” *System Sciences, 2001. Proceedings of the 34th Annual Hawaii International Conference on*, 2001, p. 5.

[12] A. Arnab and A. Hutchison, “Persistent access control: a formal model for drm,” *Proceedings of the 2007 ACM workshop on Digital Rights Management*, Alexandria, Virginia, USA: ACM, 2007, pp. 41-53; http://portal.acm.org/citation.cfm?id=1314286.

[13] C.N. Chong, “Experiments in rights control: Expression and Enforcement. PhD thesis,” Feb. 2005; http://eprints.eemcs.utwente.nl/717/.

[14] S. Michiels et al., “Towards a software architecture for DRM,” *Proceedings of the 5th ACM workshop on Digital rights management*, Alexandria, VA, USA: ACM, 2005, pp. 65-74; http://portal.acm.org/citation.cfm?id=1102559.

[15] P.A. Jamkhedkar and G.L. Heileman, “DRM as a layered system,” *Proceedings of the 4th ACM workshop on Digital rights management*, Washington DC, USA: ACM, 2004, pp. 11-21.

[16] G.L. Heileman and P.A. Jamkhedkar, “DRM interoperability analysis from the perspective of a layered framework,” *Proceedings of the 5th ACM workshop on Digital rights management*, Alexandria, VA, USA: ACM, 2005, pp. 17-26; http://portal.acm.org/citation.cfm?id=1102546.1102551.

[17] N. Rump, “Definition, Aspects, and Overview,” *Digital Rights Management*, Springer Berlin / Heidelberg, 2003, pp. 3-15.

[18] S. Guth and R. Iannella, “Open Digital Rights Language (ODRL) Version 2 Requirements,” Feb. 2005; http://odrl.net/2.0/v2req.html.

[19] “eXtensible rights Markup Language (XrML) 2.0 specifications,” 2001; http://www.xrml/org/.

[20] P.C. Kocher, “Timing Attacks on Implementations of Diffie-Hellman, RSA, DSS, and Other Systems,” *Lecture Notes in Computer Science*, vol. 1109, 1996, pp. 104-113.

[21] F. Biancuzzi, “Racing against reversers,” *SecurityFocus*, Jun. 2008; http://www.securityfocus.com/columnists/474/1.

[22] E. Diehl and T. Furon, “copyright watermark: Closing the analog hole,” *Proc. IEEE Int. Conf. Consumer Electronics*, 2003, pp. 52–53.

[23] I. Cox et al., *Digital Watermarking and Steganography*, Morgan Kaufmann, 2007.

[24] “Fighting audiovisual piracy: a good practice guide for the industry,” 2007; http://www.cnc.fr/Site/Template/T8.aspx?SELECTID=2531&ID=1661&t=1.

[25] “Decentralized DRM: Next generation of DRMs,” Jul. 2004.

[26] “OMA home”; http://www.openmobilealliance.org/.

[27] Open Mobile Alliance ltd, “DRM Architecture Version 2.0.1,” Feb. 2008.

[28] R. Iannella, “Open Digital Rights Language (ODRL) Version 1.1,” Sep. 2002; http://www.w3.org/TR/odrl/.

[29] “CMLA technical specifications,” Dec. 2005; http://www.cm-la.com/licensing/specifications.aspx.

[30] “DVB - Digital Video Broadcasting - Home”; http://www.dvb.org/.

[31] Hitachi et al., “Digital Transmission Content Protection Specification Volume 1 (Informational Version),” Oct. 2007; http://www.dtcp.com/data/info%2020071001%20DTCP%20V1%201p51.pdf.

[32] J. Kelsey, B. Schneier, and D. Wagner, “Mod n Cryptanalysis, with Applications Against RC5P and M6,” *Lecture Notes in Computer Science*, 1999, pp. 139-155.

[33] E. DIEHL, “MediaNet: A framework to unify different distribution channels,” Oct. 2004; http://www.ist-ipmedianet.org/Medianet_position_paper_final.pdf.

[34] “Content Protection & Copy Management Revision 2.0 Specification,” Feb. 2008; http://www.dvb.org/technology/standards/a094r2.1-10.CPCM.pdf.

[35] “Welcome to Coral Consortium”; http://www.coral-interop.org/.

[36] “DMP Home Page”; http://www.digital-media-project.org/.

[37] G. Fernando, T. Jacobs, and V. Swatinathan, “Project DReaM An architectural Overview,” Sep. 2005.

[38] Mc CORMAC, *European Scrambling System*, Waterford University Press, 1996.