Telecommun Syst (2010) 45: 3–20
DOI 10.1007/s11235-009-9231-4

Requirements for enforcing digital rights management in multicast content distribution

Malek Barhoush · J. William Atwood

Published online: 4 December 2009

Abstract In this paper, we have collected the requirements for Digital Rights Management from various sources, and presented them as a set of 11 requirements, associated with five categories. We discuss each requirement, provide the motivation for each entry, and illustrate how each one could be achieved. Four example commercial DRM systems are briefly explained, and the requirements that they meet are presented in tabular format. None of the example systems meet all the requirements that we have listed. The security threats that are faced by DRM systems are briefly discussed. All of the example systems are based on unicast data distribution. The use of multicast data distribution can help the source of the data and the underlying network to reduce their resource requirements when distributing high-quality content at minimum cost and delay. Up to now, there has been little motivation to use standard Internet Protocol multicast because it does not support any protection mechanisms for the delivered data. Given that significant progress has been made by other researchers in providing "secure" multicast data distribution, we explore how the use of secure multicast as a distribution technology can bring significant improvement for some requirements, while making the achievement of others more difficult. We review how the architecture of the distribution must change to permit capturing the advantages of multicast distribution while retaining as much as possible the features of unicast systems. Some open problems are identified.

Keywords Content provider (CP) · Digital rights management (DRM) · DRM agent (DA) · Digital content · End user (EU) · Intellectual property · License provider (LP) · Network service provider (NSP) · Open mobile alliance (OMA)

1 Introduction

In the early years of producing content, the relationship between the content owners and the content consumers was based on physical objects, e.g., books. The content publisher, who was responsible for publishing these books, would try to prevent consumers from compromising this service and producing illegal copies. If s/he used special paper that prevented copiers from producing (illegal) high quality books, the protection of the content owner was somehow assured [1].

The increasing reliability of the Internet and the advanced technologies used to generate digital multimedia have changed the distribution methods for multimedia content from physical forms into digital forms. From now on, we use the word content to refer to digital multimedia content (audio, video, image, e-book, etc.). This new technology draws intelligent artists' attention, converting their "tangible" [2] intellectual property into equivalent digital forms and then advertising their innovations to the whole world for little cost, especially when knowing that millions of customers can easily connect to the Internet.

Compared with a content distribution service for "tangible" intellectual property, a digital service has the potential to increase the content producer's profit. However, it has the disadvantage that a person (paying subscriber or not) can get

M. Barhoush
Department of Computer Science and Software Engineering, Concordia University, 1455 De Maisonneuve Blvd. West, Room EV 8.139, Montreal, QC H3G 1M8, Canada
e-mail: m_barho@cse.concordia.ca

J.W. Atwood (✉)
Department of Computer Science and Software Engineering, Concordia University, 1455 De Maisonneuve Blvd. West, Room EV 3.185, Montreal, QC H3G 1M8, Canada
e-mail: bill@cse.concordia.ca

a copy of the content and start to re-distribute it. This has led to the creation of Digital Rights Management (DRM) systems, which are intended to protect the content producer's rights to distribute the content, and thus retain his/her profits.

Content distribution has traditionally been based on a one-to-one relationship between the content provider and the end user. These two parties agree (implicitly or explicitly) on the mechanism(s) to be used (in the content server, on the wire, and in the receiving host) to protect the digital content from various threats, whether they come during data transmission, or after the data have arrived at the receiver.

The management of the digital rights has been the direct responsibility of the content provider. The resources of the Network Service Provider have been used solely to "move the data". The only negotiation required between the content server and the network has been to ensure that the necessary resources are available to deliver the required Quality of Service, using, for example, the Resource Reservation Protocol (RSVP) [3].

DRM systems are designed to guard intellectual property against digitally related criminal actions. DRM systems allow intellectual property owners to embed control in the delivered products to get back some of the money they spent developing it. DRM systems can be defined as cooperative and organized efforts between trusted entities and tools to achieve consistent and persistent control over digital products [4–6]. The phrase "persistent control" is used to imply control for a certain period of time, but (typically) not "forever". After a certain time, the content may become available for free, not because the owners of the copyright have given it up, but rather, because they stop enforcing it. This is because the cost of satisfying demands for new copies is more than the content producers' revenue, and they have already achieved sufficient return on their investment [2, 7, 8].

While this one-on-one content distribution model is (relatively) simple to implement, it has the disadvantage that the load on the content server increases in direct proportion to the number of end users served. As the demand for content distribution increases, the content provider becomes unable to support all potential subscribers, because they exceed his/her resource capacity (throughput bottleneck). New users who exceed the provider's capacity have to wait until the provider finishes serving old customers, which of course will cause the desired service latency of the new users to be exceeded.

Multicast as a distribution mechanism lowers the cost and latency needed to simultaneously deliver QoS real-time multimedia (digital TV channels, movies and distance learning) to multiple receivers. Multicast delivery works most efficiently in circumstances where large numbers of customers are lined up waiting for the same service. In these instances, it makes sense to replicate data packets within the network as is done in native Internet Protocol (IP) multicast.

Standard IP Multicast [9] provides no control over the end users, and no security. Hence it is not suitable for distribution scenarios where the end user is to be charged. However, considerable work has been done on multicast data protection [10–12], participant access control for end users [13] and senders [14], access policies [15], and architectures for managing secure multicasting [16–18]. Procedures for managing Quality of Service for multicast streams have been available for many years using RSVP [3]. Thus, most of the pieces are in place for establishing a "Secure Multicast" environment.

However, except for one patent [19], we have not been able to discover any published DRM system that accommodates the requirements of multicast-based systems.

Our primary goal is to formulate requirements for DRM in multicast systems. A reasonable expectation would be that these requirements would be extensions/adaptations of the requirements for DRM in unicast (one-to-one) environments. As we were not able to find any comprehensive formulation of these (unicast) requirements, the compilation of a basic list of requirements became a goal as well.

In this paper, we will explore what is being accomplished in the DRM world, extract the requirements from that environment, and then we will go through how those requirements shift from unicast to multicast. In summary, we have three contributions:

1. Collect and categorize the major requirements for DRM systems.
2. Propose the major requirements for acquiring "persistent protection" for multicast content distribution.
3. Show the comprehensive study behind these requirements.

In Sect. 2 we will discuss two related fields: secure multicast content distribution and DRM systems. In Sect. 3, we will study the DRM system in detail and discuss its requirements and the motivation behind these requirements. We will then propose the generic DRM architecture. Section 4 will show current DRM solutions and discuss four examples of DRM models and their limitations. Section 5 discusses the threat model. Section 6 browses secure multicast architectures and the need to apply DRM for that model. In Sect. 7, we will discuss the challenges for applying DRM requirements for multicast and come out with a set of new requirements. Afterward in Sect. 8, we will give a comprehensive study for the requirements introduced in Sect. 7. In Sect. 9, we offer our conclusion and suggestions for future work.

2 Related work

In this section we will explore some of the related works that demonstrate general requirements for adding data protection and access control functions for IP multicast and deploying DRM systems. The Internet Engineering Task Force (IETF) Secure Multicast (MSEC) working group proposed the Multicast Group Security Architecture (MGSA) [11] as a reference framework that provides clear-data concealment to the traditional IP multicast. MGSA introduces the need for efficient key management and distribution as well as the requirements for manipulating and carrying out access control for multicast content; however, they do not mention the requirements for accounting for the content usage. Afterwards, scalable key management protocols and schemes have been proposed by many authors. The IETF Multicast Deployment (MBONED) working group has introduced the requirements for accounting and controlling access to the IP multicast [20]; they call this "well-managed" IP multicast. None of these previous studies provide the requirements for protecting content from a hostile person who may receive the clear content legitimately.

Nickolova and Nickolova built a DRM e-learning conceptual model. They show roles' responsibilities and explore their motivation and goals, then come out with the security requirements that properly regulate them within the model [21]. Jonker and Mauw represent a DRM conceptual process model and use a structured approach to reasonably describe the security requirements for that model [22]. Arnab and Hutchison simplify the DRM problems and propose and analyse the security requirements for persistently protecting the content media within an enterprise [23]. The Open Mobile Alliance (OMA) introduced OMA DRM Requirements Candidate Version 2.1, which defines a set of DRM requirements within the OMA community [24].

In the unicast case, it is easy to establish a relationship between the seller and the buyer. That relationship is used to build an efficient solution by allowing the seller to specify some rules, and then provide a license to use the product to the buyer who satisfies the rules. DRM systems are adapted to a one-to-one relationship and thus, this is reflected in DRM requirements. The aim of our analysis is to identify the basic differences between a typical DRM model and a secure multicast model and then define the challenges that need to be mitigated in order to add DRM functionalities to the current multicast model.

3 Security requirements for DRM systems

The Internet is the global marketplace where "digital artists" can offer their digital products and gain a sufficient amount of money. DRM systems are mainly used to give content owners full jurisdiction over the content no matter where it resides, and that is what DRM is for [7]. In this section, we will present the most important DRM requirements and then we will show the rationale behind these requirements.

Accessing content media can be legal or illegal. The definition of legal access depends on the context described by content sellers. Examples of legal access to content media include: play, replay for a limited period of time or number of times and the legal redistribution of content. Examples of illegal access to content include: copying of unprotected media and modifying content data. Any access to the content that is not approved by the content sellers is illegal.

Producing a book in the traditional manner required four stages: editing, publishing, distributing and consuming. In order to make it a feasible business model, a method needed to be deployed to prevent illegal copying. Copyright law puts some regulations in place to prevent such dishonest actions. Unfortunately, this would not prevent unscrupulous customers from photocopying the book once they physically accessed it. One solution was to use special papers in the publishing phase and lower the price of the book, so these customers would not waste their time and money to produce a bad-quality replica via a photocopier machine. The protection of the book was somehow accomplished [1].

Nowadays, the digital world makes the copying of digital content media easy and perfect, which makes protecting these media more difficult. Digital rights management is a scheme designed to protect digital assets. There are four basic processes used in the DRM system: protection, distribution, management and control [25, 26]. Many researchers proposed a list of requirements in order to give the content providers the ability to acquire money for their digital intellectual property, and save the producers and consumers rights [21–23, 27–30]. We organize these requirements into five categories: access control, security, privacy, robustness and marketing:

R1 Prevent illegal access and allow legal access to valuable media.
R2 Ensure the authenticity of interacting objects [24].
R3 Regulate the legal operation of digital content, in other words, permit different authorization activities for different types of transactions [6].
R4 Ensure the integrity of digital assets [21, 28].
R5 Ensure the non-repudiation for the service [28, 30].
R6 Save the privacy of end users [27].
R7 Ensure the availability of the service [28].
R8 Reduce the damage caused by the attacker [22].
R9 Support service on demand.
R10 Ensure efficient use of content provider's resources.
R11 Allow domain access [24].

In Table 1, we map these requirements into the five categories: access control (AC), security, privacy, robustness and marketing. In the next section, we will give a justification of the elements in this table and discuss the goals behind these requirements in detail.

Table 1 DRM Requirements

Req	AC	Security	Privacy	Robustness	Marketing
R1	+				
R2		+			
R3	+				+
R4		+			
R5		+			+
R6			+		
R7				+	
R8				+	
R9					+
R10					+
R11					+

3.1 Goals behind requirements

The content owners do their best and spend a good amount of money to produce remarkable intellectual properties. They need to protect their works in order to control the use of them for a certain period of time and thereby recover the money that was spent for developing these ideas. DRM strives to achieve "persistent access control" for the content provider/owner and provides him/her the ability to regulate the digital content operations [22, 28]. This persistency is achieved by hiding the content as a first stage, then filtering the access to it as a second stage. Requirement one is about hiding the content and both requirements two and three are about filtering and organizing the content access. Hiding the content is accomplished by making the following sub-requirements valid:

R1.A: Prevent the action of capturing clear content in the distribution path (Access control requirement).
R1.B: Prevent the action of stealing clear content data when it is hosted at end users' machine (Access control requirement).

To allow only legitimate customers to utilize the service, a demand for identifying and authenticating the customers is needed. Requirement two is to ensure the authenticity of the following:

R2.A: Digital assets.
R2.B: Sender entity.
R2.C: Receiver entity.

This involves clarifying, ensuring and assessing the truth of any declaration sourced by a valid entity (content sender, content consumer and the digital assets), we consider it as security requirement. To start with, consumers want to ensure the identity and authenticity of the sender(s) before receiving any data, and this is useful for protecting customers from sender spoofing. As well, a sender requires that each valid customer be identified and authenticated before s/he is authorized to use the product, and this will help for billing issues. Finally, the content consumer hopes to authenticate the product itself before s/he starts using it. This process helps him/her to avoid any harm that could be sourced from unknown content providers or products.

Requirement three introduces the need for managing different business models and accounting for the content usage, that is why we classify this requirement as both access control and marketing. It is further subdivided into the following sub-goals:

R3.A Content owners need to specify their content-usage policies (rights/licenses) (Marketing requirement) [22, 31].
R3.B Content usage specifications need to be protected and distributed to their appropriate destination (Access control requirement) [22].
R3.C Specified usage activities need to be enforced (Access control requirement) [21, 32].

Requirement four ensures that any digital assets used in the context of content distribution have not been changed in the path from content providers to content consumer, and it takes two flavors: checking the digital assets' integrity in the distribution path and giving a promise to the Stakeholders that the digital asset will not be changed at the point when the end user can access it; we considered it a security requirement, because if the integrity of the assets is violated, then it could introduce a hole in the content distribution model.

Requirement five is to ensure the non-repudiation action for the requesting process, which is an important security service that avoids any tensions that could happen between content seller chain and the end users. There should be evidence of selling/buying the product for both sellers and buyers. The seller needs this service to prevent the buyer from denying using the service and not paying the fees. At the same time, if user protection is broken because of a delibrate security hole embeded inside a product sold by the seller, then this offensive action should not be denied by the seller, this is a kind of security requirement. To conclude, both the sellers and the buyers should be responsible for their activities. In another view, if the access to the service is granted, then this action should not be repudiated. We consider this requirement as both security and marketing requirement.

The sixth requirement affirms anonymity to end users by preventing unauthorized entities from accessing users' private information such as name, address, date of birth, credit card number, and so on. This information could help the attacker to gain access to transactions that belong to someone else. It is a privacy requirement.

Requirement seven is important for all parties. It concerns keeping the product service, the users can receive the digital product and its license if they are eligible without any blocking, which means that denial of service (DOS) attacks must necessarily be eliminated. We considered it as a robustness requirement.

Requirement eight tries to reduce or mitigate the unsatisfactory effects of service attacks. It tries to find a means of defense against intentional irresponsible actions and return the system to the previous stable state, it is a robustness requirement. We further subdivide it into three lines of defense.

R8.A Prevent "break-once, break everywhere" (BOBE) [2, 22].
R8.B Detect and fence the cause of illegal content distribution.
R8.C Revise the protection engine once it has been compromised [22, 33, 34].

Requirement nine endows the DRM system with flexibility; once the users choose the time frame for enabling legal operations on the multimedia content, they should able to do that [22]. Somehow, it is related to the availability requirement, R7. This requirement is important for marketing issues, because if the service is motivating users, then the service provider is successfully marketing her/his product.

Requirement ten seeks the efficient usage of the content providers' resources. It contradicts requirement nine, which demands reserving fixed resources for each individual user. Because of the fact that senders have limited resources, they can serve only a limited number of users. Therefore the flexibility desired by requirement nine is influenced by the efficiency desired by requirement ten. Requirement 10 is again a marketing requirement.

Requirement eleven gives the ability to each customer to use the same content on a limited number of devices s/he owns. It is preferable that deploying any technique to achieve this requirement not hurt any of the previous requirements. This requirement attracts a customer to use this service, therefore it is a marketing requirement.

3.2 Generic form of the DRM solution

In this section we will show some methods used for achieving the requirements mentioned in the previous section.

A digital rights management system (DRM) divides the world into two sets: Allowed and Prevented. Prevented set contains all entities (machines/customers) that are not allowed to use a specific product. In contrast, Allowed set contains only entities that are granted authorized operations on a specific product under the terms and conditions stated by the product owner.

To achieve a "persistent access control" for the content usage, it is sufficiently recomended to consistently enforce the first three requirements (R1, R2, R3). The first requirement is divided into two sub-goals. The first sub-goal R1.A insists on resisting the action of capturing the content in the distribution path. The distribution path in this context is the Internet or public networks. This could be achieved by using cryptographic techniques to obscure content data. This could be achieved by providing the means to the Content provider to sufficiently establish an individual secure channel between him/her and each individual user, and that secure channel is guaranteed by encrypting the content and keeping the encryption keys out of the hands of ineligible users. Giving the encryption key only to those users who can provide sufficient funds gives them the right to use those contents. This mechanism provides clear-content access control at the network level [28, 35, 36].

Now, once a dishonest legitimate user gets the encryption key, s/he may introduce a dangerous security hole by publishing that key. The second sub-goal R1.B is to prevent any dishonest users from re-distributing either the secret keys or the clear contents once they enter the user's machine, in other words, keep disingenuous persons away. This can be achieved by using hardware or software tamper resistance (trusted component) as a base for hiding the key or the technologies used to protect digital multimedia system, "security by obscurity" [37]. Deploying these trusted components could prevent legal users from holding encryption keys or knowing other related secrets, as a consequence, they could not directly access the clear content [38]. This mechanism gives the meaning of protection persistency anywhere the content media may be [28], but it does not give any hint as to how long this persistency could last.

DRM systems physically separate the content media into two components: the protected content media and the media license. The license gives the customer the legitimacy to access and use a specific content medium. The license comprises permissions and constraints for using content, which reflects the usage policy [25]. Licenses are bound to their corresponding content media by attaching sufficient information (metadata) to the protected content in order to guide the consuming devices to the location where the license can be acquired. In this case, DRM designers can authorize legitimate users in a distributed and scalable manner.

Once the protection is achieved, the whole world is placed in the Prevented set. To enter paying customers into the Allowed set, an authorization means must be defined. Before authorizing selected users to render the content and moving them from the Prevented set into Allowed set, they are supposed to be identified and authenticated, if only for billing purposes. In some cases, identifying all parties is essential. As an example for achieving Requirement two, a PKI infrastructure could be used. Each entity owns a unique public/private key and a digital certificate, which is published by a known certificate authority. Each entity can mutually authenticate each other by using the other public key to encrypt part of the exchanged messages, thereby only one target can decrypt these parts. To authenticate Digital assets, a signed secured hash for those assets is guaranteed to achieve its authenticity when a receiver can successfully check the signature of that hash.

The need for managing the distribution of the authorization means is required. A DRM system achieves this management by producing a digital license for each protected content medium. The digital license enables the rendering process for the protected content; therefore, since digital data are easily stolen, licenses should also be protected. This is the main subject of the third requirement. Let us go through the three sub-requirements of requirement three.

Sub-requirement R3.A could be achieved by providing a suitable medium for content providers to express their policies in order to authorize legitimate customers to use their content media. A rights expression language (REL) allows those providers to state their access requirements, permissions and constraints, as well as, to include encryption key(s) in the license file. Examples of REL are extensible markup language (XML) and open digital rights language (ODRL). Creating different licenses for the same content media allows different business models (e.g., preview for free, pay per view, rent, subscribe), which depend on end user's payments and content sellers' policies. Because the CP policies grant rights to end users when they are promising to pay and thus invites them to the allowed set, policy links the money into the acquired rights. An REL needs to be: comprehensive, generic and precise [7, 28].

One way to achieve sub-requirement R3.B is to protect and wrap sensitive parts of the license into a container [36], and this will prevent unauthorized customers from using the license if they do not have the right means to do so.

The next step to complete these authorization issues, it is recommended that consuming devices works with both protected content and a suitable license in order to give the rendering application the ability to present the content media. One way to do that is using a DRM mediator within a consuming device, this mediator prevents customers from directly accessing the content media and enforces and controls legal operations upon digital contents as is set out in the license, and this is what sub-requirement R3.C states.

Requirement four demands verifying the content integrity before starting to use it. One means to permit the integrity checking, the sender may generate the content message digits, for instance using MD5 or secure hash using SSH-2, sign it and then attach it along with the content. Then on the other side, the receiver generates the content message digits or secure hash and then verifies the result with the received value. Theoretically, everyone has a distinctive signature, then the signing process is a proof of the sender's authenticity and the hash is a proof of message integrity. Checking the integrity requirement protects entities from unauthorized modification of the content.

The signing process is a proof that only the sender did the sending process, assuming that each server has its own private key and there is no real attack against the PKI infrastructure. This is what requirement five is about. We need a mechanism to prevent the customer from denying the receiving services, as well as the sender can not deny the sending process.

The sixth requirement suggests that DRM should stop unauthorized entities from accessing private information that belongs to the legal customers. This gives the confidence to end users that their private information will not be misused. Unfortunately, this requirement is not fully achieved by current DRM systems, because they depend on installing software that does active protection on the client side, which may work as a rootkit [39]. A customer needs to be assured that the application software used to render the content as well as the entities that play the role of content providers are trusted and not doing something harmful toward her/him.

Requirement seven suggests maintaining the availability of the DRM system, which could be achieved by replicating the roles of both content provider and license issuer and maintaining their consistency. The super-distribution process [26, 36, 40] helps to sustain the availability of the content distributor by releasing him from delivering the content media, and thus, saving the DRM system resources.

Requirement eight tries to mitigate the effect of compromising the content distribution service. It takes three actions, which are stated in three sub-requirements. Sub-requirement R8.A is a prevention strategy and it is used as a second line of defense, hiding the content is the first line of defense. Its suggestion is to quarantine compromised consuming devices away from healthy ones. This is achieved by limiting the damage caused by the attacker to a narrow compass. This could be accomplished by using different keys to protect multiple instances of the same product, so compromising one copy or exposing one decryption key will not affect others. Microsoft defines an individualization concept for this purpose. It divides the workflow for the content publication lifecycle into components, and then binds one of these components into requesting hardware, which makes that application instance unique to one consuming device. Thus, any compromising for one instance does not affect other instances [2, 35].

Sub-requirement R8.B is a kind of detection strategy and it is considered as a third line of defense. It is achieved by adding a unique mark, a watermark or a fingerprint, to the content media before protecting and delivering it to the end user. This mark is supposed to guide the DRM monitor to recognize the source of distribution after the fact of illegal distribution into the "Darknet" [2, 41]. Watermark/fingerprint insertion requires much processing time, so, for efficient delivery, this insertion process is normally done offline before hiding the content.

Sub-requirement R8.C is a reactive mechanism, it suggests that the compromised protection engine should be changed once it has been compromised. Bar-El realized that hacking any fixed scheme is a matter of time, therefore replacing a compromised scheme with a healthy one is a promosing solution. Bar-El proposed the challenges that face this replacing and defined the guidelines to solve those challenges [2, 33]. The feasibility of this strategy depends on successful attack rate, the time needed to replace the old protection engine, and the provider's capacity to afford simultaneous changing. Those factors determine the scalability of this renewability mechanism, so if the renewability is done regularly for a certain short periods, the extensive overhead would be unaffordable.

Requirement nine (support service on demand) could be achieved by allocating sufficient resources for each client. This reflects that the number of concurrent connections between content provider and end users is limited to the service provider's capacity. Because each request is maintained by a single connection, a service provider can serve a limited number of customers. The efficiency of the system depends on the service type and content media size. Low-quality files, a small size each, need a small time to be downloaded, so the delay time for a customer to wait when the number of concurrent connections exceeds the service provider capacity is small. In contrast, the delay time for video files is large for the same scenario. This is why the OMA community is able to serve a large number of simultaneous connections: end users will not complain for small delay in services such as rings or audio songs.

Requirement ten is achieved by adding an intermediate cache near a collection of end users. This cache is used to save favorite contents, so end users can efficiently retrieve cached media. This will lessen the extensive load on the content provider, improve the latency time and minimize data exchange in the network level. Some DRM systems introduce a peer-to-peer super-distribution service [26, 36, 40], which will allow the end user to forward the content to another peer user without providing her/him the license, and then the new peer user needs to acquire a new license. This mechanism will reduce the competition for the sources' resources.

Requirement eleven is achieved by enabling the transfer of a license to a limited number of devices within a domain, so if one user owns more than one device s/he can declare a domain collecting these devices, then s/he can view her/his contents within these devices using the same license [42].

3.3 DRM generic architecture

The DRM general architecture consists of three players: content provider (CP), license provider (LP) and end user (EU), see Fig. 1. The CP mainly is responsible for generating, hiding content media and attaching some meta-data along with each hidden content. The meta-data guides the consuming device to the location of the LP, i.e., where to acquire a license. The CP provides the LP with corresponding content encryption keys (CEK). The LP is mainly responsible for creating permissions (licenses), which include terms and conditions, as well as the CEK for enabling the consuming device to expose the corresponding hidden content. EU downloads the hidden content via local software called a DRM agent (DA) that is designed to enforce usage policies. The DA extracts the information pointing to the LP from the meta-data, negotiates with the LP for providing licenses according to user's payment amount, downloads the license, checks the integrity and the validity of the license, interprets the license, extracts the CEK and enforces the terms and conditions [26, 41–43].

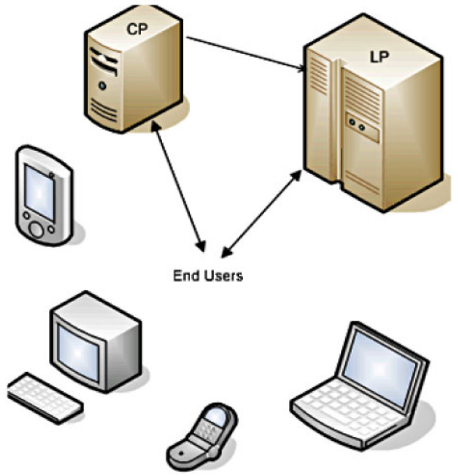

Fig. 1 DRM Generic interactions

In most of the DRM systems, hidden contents can be publicly reached either from the CP or via another peer device (super-distribution). However, the license file that allows the completion of the rendering process for any distinct content must be paid for. Therefore, controlling and managing the license helps the content owners to make money.

4 Current DRM technologies and their limitations

DRM technology are deployed in three levels (application, operating system and hardware [7]). We will talk about two successful DRM products belonging to Microsoft, which deploy DRM in operating system and application level: Windows media rights manager (WMRM) and the successor Windows Rights Management Services (RMS). Then we will talk about the Open Mobile Alliance (OMA), which uses DRM application and hardware level. We will end our discussion with the ISMACryp framework, which results from adding DRM to the Internet Streaming Media

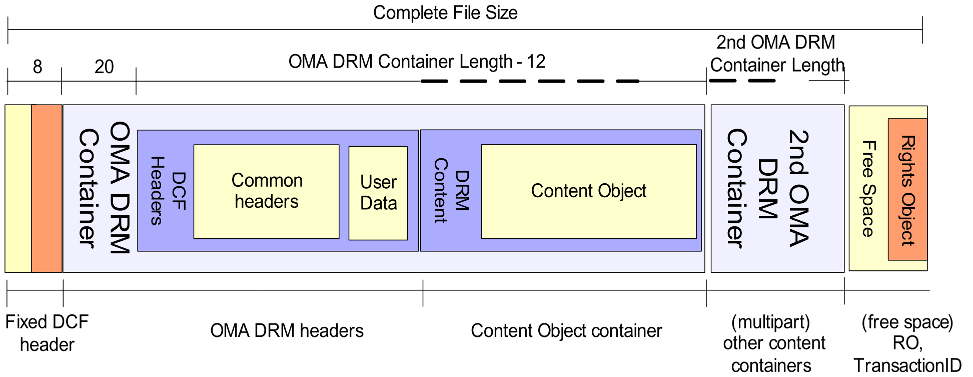

Fig. 2 Data Content Format structure [36, 43]

Alliance (ISMA) model. ISMACryp is built in the application layer [44]. We will use labels (R1…R11) beside each process activity to indicate satisfaction of that requirement.

WMRM provides an economical and feasible solution for hiding the digital media for a while; it does not need any special hardware to hide the content. In contrast, RMS may need such hardware. Both technologies (we will call them Microsoft DRM or MSDRM) support multiple business models.

Microsoft released operating systems that allow Microsoft DRM to be run through a variety of devices such as personal computers, notebooks, PDAs, smart-phones and pocket PC [45, 46]. WMRM as well as RMS are an end-to-end DRM solution. They support cryptography mechanism for secure distribution (R1.A) and a secure environment via individualization and revocation process (R8.A) as well as secure path to the hardware driver (R1.B). They support machine authentication [47, 48]. There is no provided information that tells us that WMRM supports user authentication but RMS does (R2.C) [49, 50].

For each enterprise, there is a certified RMS server used for registration purposes, the RMS system considers this server to be a root server. The registering RMS server signs up each client's device; it has the chance to register other servers (R2.B). In the Microsoft RMS system, each client needs to use a DRM controller (R3) and his account certification in order to enable DRM. RMS may authenticate enterprise internal users as well as external users (users who do not belong to the same enterprise) as long as they use either active directory server or .NET Passport account (R2.C) [50].

MSDRM uses the "individualization" technique, which generates a unique instance of the software player, and binds each instance to a specific customer machine, therefore, each player is supposed to work only on a specific machine. It also supports revocation service as counterattack if individualized software instance is compromised (R8.A). More information on these two services is available on the Microsoft website [51].

The WMRM player is software and it is susceptible to modification or replacement attacks. These attacks are achieved by obstructing or modifying the enforcement part of the rendering code with an attacker-made code, and thus bypassing the checking points. Another attack was created by one software cracker. He analyzed the WMRM code and then produced a tool called "FreeMe", which tracks the location of encryption keys located inside the blackbox file (used to hide these keys) and then exposed hidden media files [22, 52]. WMRM does not provide real privacy preservation for end users; neither does RMS (R6). There is no reported attack against the RMS system but the behavior of the system indicates that it is susceptible to a software reverse engineering attack.

The Open mobile alliance DRM-2 (OMA-DRM-2) [31, 36] is a specification and standards designed for enabling the control of digital services on different mobile phones and personal players. OMA DRM2 architecture has three major components: content issuer (CI), rights issuer (RI) and a DRM agent (DA). The CI generates a content encryption key (CEK) either for each individual content medium or multiple contents. It may encrypt selective contents and then package each of them in a secure container; this container is grouped and packaged into the DRM Content Format (DCF) (R1.A). The DCF may contain more than one container, Fig. 2 shows the DCF structure [36] (R4). OMA-DRM-2 supports a small-size content DCF (picture, ring and small messages) as well as a large-size (audio and video content), they call it Packetized-DCF (PDCF). The CI negotiates the rules and constraints for DCF usage with LI (R3.A). CI delivers DCF to customer machine via various transport mechanisms, s/he does not need to use a secure connection since the DFC is already secured (R1.A).

When a customer browses a digital catalog, selects interesting media to play, reads and agrees on terms and conditions and the price for consuming that content, the SIM card that is attached to the mobile phone authenticates the user (R2.C). Afterward, the DA, which plays the tamper resistant role residing in a mobile station, requests the protected content. DA downloads a DCF, checks its integrity (R4) and extracts the information that triggers it to send a rights request to the LP. When it receives the rights object (RO), it verifies the authenticity of RI and RO as well as RO's integrity, all authentication activities happen through rights object acquisition protocol (ROAP) (R2) [31]. By then, DA extracts the keys (KEK) from RO and decrypts the protected contents within DCF/PDCF.

The LP creates a suitable rights object (RO) for each DCF. The RO works like a license. When the DA requests an RO, the LP authenticates the DA and protects the RO, which is achieved by encrypting the part containing KEK with target DA's public key (R1.A, R2.C) and then signing the RO (R2.B). This means if an adversary accesses this RO, s/he can not access the KEK because of not having the corresponding private key (R8.A). The RO is an XML file containing DCF encryption keys and expresses the rules and constraints for using the DCF as being expressed by the CI (R3.C).

The DA enables the content rendering process and controls its usage rules. The DA is a trusted component in the mobile phone (R1.B); it has a unique public/private key and a certificate, which helps the LP to authenticate the DA (R2.C). The DA is designed in a way that it should receive both DCF/PDCF and the associated rights object in order to render protected content; it checks and governs the treatment of the DRM content by enforcing the rights stated in the rights object (R3.C). The keys in the Rights object are encrypted with the DA's public key. This process binds the rights object to a specific DRM agent and only that target agent can expand the encryption key out of the rights object (R8.A). It is possible that DA redistributes a protected container to another friend's machine (super-distribution), the receiving machine's DA will start a new RO acquisition process for the received DCF. This super-distribution decreases the extensive overload on CP (R10) and improves the availability of the service (R7).

OMA DRM-2 supports the domain concept, which allows the sharing of RO to a group of domain registered devices; this allows them to veiw the same contents using the shared RO (R11).

Mobile phones have unique embedded proprietary hardware specifications, and each user has to use a special smart card, which supports the device with an address number, therefore, user/device identification, authentication and payment are reliable (R2.C). Embedded hardware works as a tamper-resistant hardware and it provides the trust to the LP [53, 54].

Internet Stream Media Alliance Encryption and Authentication (ISMACryp) is successfully used as a business model by generating a controllable streamed service for high quality media content such as video and audio. The main purpose of this service is to preserve interoperability, especially when DRM is applied to the ISMA scheme. ISMACryp is being built in the application layer [44]. The ISMA architecture consists of four parties: the mastering, key/license MGT, the sender and the receiver [55, 56], see Fig. 3.

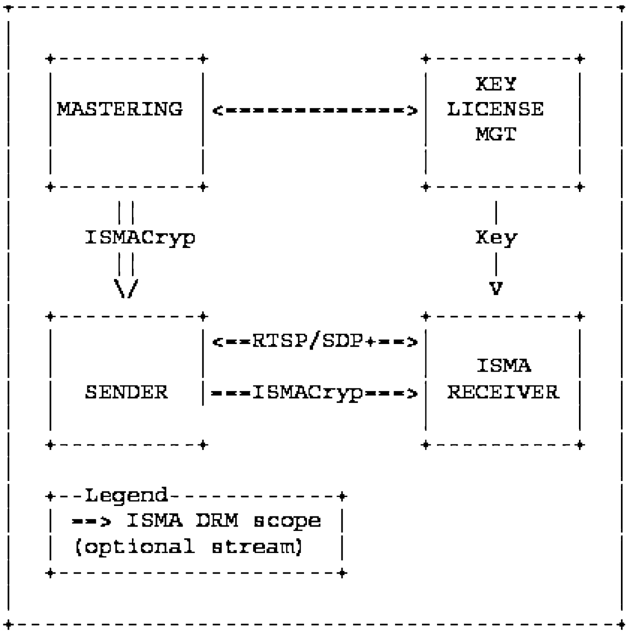

Fig. 3 ISMA DRM Architecture [55]

The mastering is responsible for equiping the content for distribution; it has the option to encrypt the media content and specify the usage rights, which helps it to work as a clerk (R3.A), or it may provide the encryption key to the sender and the sender will do the encryption part. ISMA uses "Advanced Encryption Standard Counter Mode" (AES-CTR) for protecting the content media [57] (R1.A). Mastering may get the encryption key from the key/license MGT or provide the key whenever it is needed. In the scenario where the encrypting of the media is done by the mastering entity, the sender is not aware of the encryption key; it has no job but to send the protected media whenever it is needed to the customers. Finally, mastering is responsible for advertising for the content media.

The sender is responsible either to encrypt the content media or receive the protected media from the mastering entity, and then stream it to the receiver (R1.A); the distribution mechanism for the streamed protected content is the real-time transport protocol (RTP) [44, 56]. The sender has the option to predict the encryption key by following the same procedures that are given to a user by the key/license MGT, and then generate the protected media. The media could be saved in a file before it is being streamed or streamed directly from the sender.

The key/license MGT is responsible for generating suitable licenses according to the selected business model; the license authorizes ISMA users to use the protected content media. It contains the two main components: the decryption key and the usage rights (R3.A). For interoperability issues, ISMA tries to include all type of licensing schemes. If the responsibility for creating the key encryption is on the key/license MGT, then it may generate that key depending on some properties of the receiving entity, which are used for authenticating the receiver (R2.C). The ISMACryp uses the secure real-time transport protocol to authenticate protected content (R2) and uses existing key management standards, which provides the flexibility for the content provider to chose which key management he is going to use [55].

The ISMA framework supports three types of receivers: ISMA-only-receiver, MPEG-receiver and IPMP-X receiver [55]. The first type represents the receivers who play streamed MPEG media, the second represents the receiver that can play streamed/files of type MPEG-4, and the last one represents the receiver that can parse and process Intellectual Property Management and Protection Extension (IPMP-X) format. The rendering software on the receiver side is responsible for contacting the sender and key/license MGT, authenticating both of them (R2.B), acquiring and authenticating the license. ISMACryp uses the secure real-time protocol (SRTP) for integrity checking (R4), accessing the proper decryption key for the protected content media and enforcing the usage rights (R3.C). The receiver's rendering software has the option to decrypt, authenticate and check the validity of the control flow between the sender and the receiver (R2.A). The receiver's rendering software enforces the usage rights [55]. In the ISMACryp specification, the protected media are decrypted only just before they are decoded by the rendering software (R1.B) [44].

ISMACryp supports super-distribution by providing the ability to OMA-DRM2 compliant devices to store streamed media into DFC/FDFC file format. Then, the compliant device can super-distribute the content to other friends' devices. That will increase the availability of the service as well as decrease the load on the sender server (R7, R10). Again, those devices should acquire a license in order to play forwarded content media [55].

It is known that using a public key infrastructure for hiding data is robust, but it is inefficient, especially to protect real-time streaming media. For efficiency purpose (R10), the ISMACryp framework uses a symmetric algorithm for data encryption (R1), authentication (R2), and integrity purposes (R4). ISMACryp has the option to change the mechanisms used to for encryption, authenticating and checking the validity of the message sent via this system (R8.C) [55].

Table 2 addresses the availability of the previous requirements in each mentioned technology.

5 Threats

Information security developers are concerned about countering threats. In this section, we will explore threats that content distribution services are facing. Since DRM developers have found that the easiest way to develop a DRM system is to consider that the end users's devices are trusted, therefore the most important role in the DRM system is the role of DRM agent (DA), which enforces the compliance to the content owner(s)/publisher(s) definition of legal activities on the content media. The DA controls the use of protected content media by burying secret keys used to decrypt that content, i.e., the DA works to provide piracy protection. The software attackers try to break the DA by exposing these keys and gain access to the clear content media. The DRM systems seem to be reasonably stong, however they face the following threats [39, 58, 59]:

(A) All threats that are faced by transmitting data through insecure channels are applied, e.g., data eavesdropping, man-in-the-middle and modification attack.
(B) Reverse engineering of the DRM mediator software in order to deduce the location where the sensitive data reside.
(C) Monitoring the system behavior at runtime in order to observe the data changes in memory locations, so the attacker can predict sensitive secrets hidden inside the memory.
(D) Modifying the DRM software that is used to enforce usage policies, and then bypassing the enforcement point.
(E) Modifying the license in such a way as to allow customers to use fake rights.
(F) General operating systems suffer from many security holes. In MS-Windows OS, attackers have physical access to the machine's memory and disk, and are able to hide their spyware and dangerous files [58], which helps the attacker to spy without being caught.
(G) DRM application may enable private information spying [39].

Deploying a DRM software solution within generic machines is viable and cost effective, but it is susceptible to various attacks and it degrades the system performance. Normally, because DRM processes reside in customer-machines' memory, attackers can physical access to these memories, thereby, they can reverse engineer, disassemble and decompile binary codes inside these memories and then extract sensitive information used for content media hiding. Or, they can dynamically monitor process excution and follow the pointers to the location of secret keys [60]. In addition, a software solution may carry a serious attack against a customer, specially when it works as a virus spying for private information [39].

We consider DRM hardware solution as a black box, which hides sensitive secrets and prevents them from being released. This depends on the fact that the attacker has little knowledge about that box's internal structure, and the existing tools' capabilities are too limited to catch much useful

Table 2 DRM Systems

Req	WMRM	RMS	OMA	ISMA
R1.A	Secure Channel	Secure Channel	No secure channel	AES-CTR [55]
R1.B	Securing rendering path	Securing rendering path	HW & SW tamper resistance	Client viewer
R2.A	Certificate, watermark & fingerprint	?	ROAP authenticates content	Message authentication code (MAC) [55]
R2.B	Not mentioned	Certificate	ROAP & LI signs RO	MAC
R2.C	No user authentication	MS active directory	ROAP, PKI infrastructure, SIM & USIM cards	MAC
R3	Licenses	Licenses	RO & DA	Licenses
R4	Secure hashing	SHA-1	ROAP checks FDC/PDFC integrity [31]	SRTP message authentication [55]
R5	Not supported	Not supported	Not supported	Not supported
R6	Not supported	Not supported	Not supported	Not supported
R7	Service replication	Service replication	Super-distribution	Super-distribution
R8.A	Individualization	Individualization	Encrypt KEK by DA's public key	Not supported
R8.B	Offline search, watermark	Offline search	Watermark	?
R9	VOD	VOD	Service on demand	?
R10	P2P distribution	P2P distribution	Super-distribution	Symmetric cryptography & Super-distribution
R11	Support multiple media format	Support multiple media format	Domain registration	?

Table 3 DRM Software and Hardware

	DRM SW solution	DRM HW solution
Attack	Easy: tools are available	Hard: need special tools
Fix	Reinstall new version	Replace or repatch HW component
Other problems	Needs more computational cost	Power surges & costly installation

information. Therefore, tampering with such hardware is too limited. This makes it a promising solution for future trusted computing [61]. Unfortunately, this solution is not economically feasible for existing PCs. Table 3 shows a comparison between DRM software and hardware [59].

6 The need for DRM multicast content distribution

We notice in the generic DRM architecture that it has at least three actors: CP, LP and EU, and the network providers only provide a vehicle of communication between them. The mentioned DRM requirements are feasibly efficient to deploy service on demand, as long as the network speed is high and exceeds the expectation of user demands. This allows feasible and economic digital delivery relative to the beginning Internet age where the network cost was far greater than the value of digital content itself. Now, the network becomes a feasible medium to move high quality data such as video.

The demand for this activity is going up day by day. As long as the network is sufficient and exceeds the expectation of the end users, and the capacity of the provider is sufficient, the DRM model is economically feasible. However, it is based on a one-to-one relationship. As the demands of end users grow and the popularity of this service delivery increases, we will reach a point where the expectations of the end users are going far beyond network capacity, e.g., everybody wants a hundred Megabits network speed, and the network providers will not be able to support high speed delivery to thousands of users simultaneously. If we could schedule the delivery of these highly demanded services, then a promising solution is to shift from unicast to a multicast delivery mechanism.

The advantages of using IP multicast did not motivate content providers to use it as a distribution mechanism because it only offers free join and does not monitor the sending process or restrict the receiving process, in other words, the content providers cannot control their contents' distribution, and thus, they can not recover the money they spent on producing their valuable contents. Let us study the distinctive properties (factors) that introduce the scalability to the IP multicast scheme:

F1 Separation of concern.

When a problem becomes more complex, the most effective way to deal with it is to divide it into sub-problems, solve these sub-problems and then gather the sub-solutions. When a sub-problem is assigned to an individual role, this procedure is called "separation of concern" in the software engineering process. DRM developers use this concept in their design; they physically separate the content media from the authorization mechanism in a distributed manner [62]. The IP multicast generic architecture consists of three roles: the sender, interconnecting networks (routers) and receivers. IP multicast separates the sending process from the distribution process, the distribution process from delivering process and the delivering process from the registration process; the sending process is done by the sender, the distribution tree is generated by cooperative intermediate routers, the registration process as well as the delivering process is managed by multicast-enabled routers at the edge of the network. The sender of group data is concerned about creating the group and sending the data that belong to those group members (group data). The interconnecting networks, routers, are concerned with building the multicast distribution tree, copying, forwarding and delivering the group data at the network level. Routers in the multicast case keep more information than in the unicast case, e.g. sender address, group address and output port [63]. Receivers show their interest to receive/stop-receiving the transmitted group data by sending (IGMP/MLD) join/leave requests. Thus those parties have to collaborate with each other to perform the multicast distribution. In general, for a complicated interactions, it is advised to divide it into a set of cooperating interactions and apply the separation of concern concept, which introduces the scalability, flexibility and simplicity to the solution.

F2 Resource reduction.

The number of packets travelling inside the network using the IP multicast model is smaller than in the unicast case. Suppose we want to send a data file to a group of users; having created a shared distribution tree for each sender in a multicast case will definitely improve network bandwidth, since sending N copies of packets to N customers in the unicast case is replaced by one copy of packets using the shared tree. In the multicast model, the multicast enabled routers replicate packets belonging to a group of users and forward them through their appropriate ports. In the unicast case, if a router happens to be in the path between the sender and N receivers, it will forward the same packet N times. The multicast mechanism provides a good solution for saving network bandwidth and decreasing the data traffic, assuming we have a large enough set of receivers and the control trafic is small relative to the data trafic.

F3 Better response time.

Interaction between roles affects the response time and that is what the separation of concern concept asserts. The sender should not directly manage the customers; doing that will affect the scalability of multicast sessions. S/He may not directly be responsible for authenticating, authorizing or accounting for them. This will lessen the interactions between a sender and all receivers. Intermediate routers manage the join and leave process for each user, so the number of control messages between the receivers and a sender is minimized. This improves the response time for the receiving service and improves the network bandwidth as well.

Content providers/owners insist on securing the multicast distribution model to control the use of intellectual property. This kind of protection needs to be valid for a certain period of time to recover the funds that were spent to develop such ideas. If we could not develop a protection mechanism for multicast, then no company will develop intellectual properties, because they would spend a lot of money for producing and developing these properties and not be able to make any profit from them.

The Internet Engineering Task Force (IETF) Multicast Security (MSEC) working group added two new players (the Group Controller/Key Server (GCKS) and the policy server (PS)) to the conventional IP multicast in order to introduce the confidentiality for transmitted data in the network layer [11].

In Fig. 4, the left column shows a simple structure, with a PS, a GCKS, the sender to a group, and one of the receivers. This structure suffices for small groups. When the group is larger (perhaps spanning more than one administrative region), the PS and the GCKS may be replicated, as is shown in the right column, and an individual receiver will connect to its local GCKS. Further details are available in [11].

This PS/GCKS model provides an example of the separation of concern concept in addition to the flexibility and simplicity of IP multicast design model. The GCKS is the core entity in the new architecture; along with the policy server, it is the manager of the group and is concerned with maintaining the confidentiality of the data being sent to all group members. Data confidentiality is achieved by protecting the

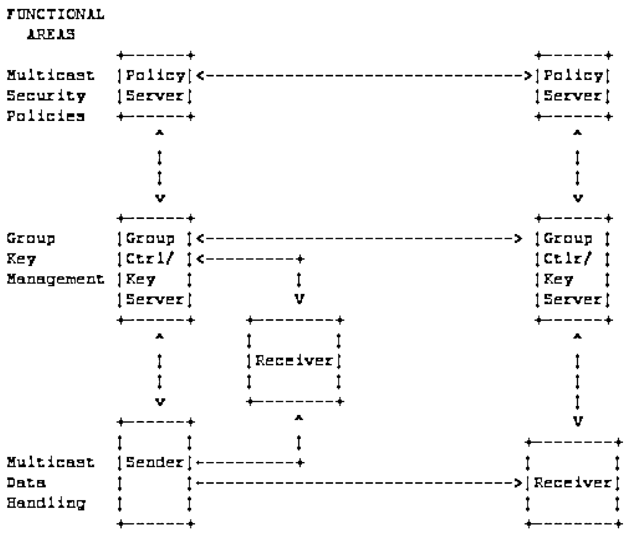

Fig. 4 Multicast Group Security Architecture [11]

data before making them available [11]. Many proposals are presented for improving the scalability of the manager task.

As the delivery of high quality multimedia content becomes faster, because of the increasing development of networking technologies, the location or distance of target consumers ceases to be of concern. This increases the potential size of the target audience for a specific content stream. If "video on demand" (VOD) is required, then each request is likely to occur at a different time, and therefore must be managed separately. The content provider/owner will take the responsibility for managing a session, and maintaining the detailed corresponding records (session keys, accounting information, etc.). As the total number of participants in a session increases, the load on the content server increases in proportion. Eventually, as noted above, a point may be reached where the content server is unable to sustain the necessary flows, and it becomes useful to consider relaxing the "on-demand" requirement, in favor of the requirement for "efficient delivery", by using scheduled delivery and multicast data transmission. However, the supervisory relationship that exists between Content Provider and End User in the first case is no longer present in the second case, because to maintain it would hurt the scalability of the multicast model (F1, F2, F3). In this case, an efficient solution is to give this supervisory responsibility to intermediate proxies, and offer the content providers a limited summary. This was proposed by Islam and Atwood [64], see Fig. 5.

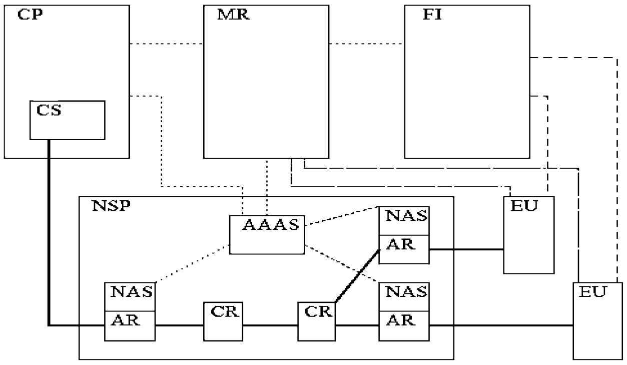

Fig. 5 Multicast Security Architecture [64]

The increasing development of the network technologies; this will remove any need to consider the location or distance of target consumers. If we consider video on demand (VOD) a small transaction between content provider and a customer, the content provider/owner will take the responsibility of managing and maintaining the detailed corresponding records. Multicast content distribution (ex. scheduled program), which consists of multiple instance of unicast connections (distribution tree), is a group of transactions, and the supervision that exists between CP and EU in unicast case is no longer there in multicast case, because that will hurt the scalability of the multicast model (F1, F2, F3). Then, the efficient solution is to give this responsibility to intermediate proxies and provide the content providers with a limited summary, and that what Islam and Atwood proposed [64], see Fig. 5.

The Multicast Security Architecture (MSA) consists of five players: content providers, merchant, financial institution, network service provider, and end users. MSA prevents unauthorized people from accessing the clear-contents flow in the network level by encrypting the data flow, and gives the decryption keys to the authorized customers. Again, this will not prevent an authorized skilled attacker from illegally redistributing these data once the protection is removed. This leads us to the point of announcing the need to deploy the existing DRM standards in secure multicast distribution, while taking care of the multicast scalability consideration. If we adopt DRM standards as an underlying layer for building a new multicast delivery model, this model will break, because existing implementations for DRM systems involve direct communication from content provider to end users, and it is no longer there in the multicast case. Therefore, the implementation model has to change to meet the scalability requirement of multicast, and that reflects on the DRM requirements also.

7 Requirements to add DRM for multicast distribution

Current DRM requirements are feasible for DRM systems up to the point where the server capacity cannot satisfy users' demands. By the time we reach the situation where the size of their demands goes beyond the server's capacity, we probably have enough customers to drop the "service on demand" requirement and replace it with an offering of scheduled services. This enables the content owner to upgrade the DRM performance, and therefore, serve more customers. However, it impacts the solutions to the DRM requirements, because multicast as a distribution mechanism truncates some DRM requirements and improves some others as well as produces new challenges in order to mitigate the others. We summarize the following requirements that need to be adjusted in order to enable DRM in the multicast model (DRMM), then we will discuss these requirements in detail.

R1 Prevent illegal access and allow legal access to valuable media.

This requirement is to attain "remote control" along with the "Persistent Access Control" [28] on multicast contents distribution, we need the following sub-requirements to be valid:

R1.A Prevent the action of capturing the content in the network level.

This requirement is being achieved by hiding the content using cryptography techniques and key management protocols, and then allowing the legal customers to use the content by giving them the means to unhide the hidden content.

R1.B Prevent the action of illegal copying of the clear content from customers' machine.

We need to build a DRM software mediator that controls the use of content and prevents end users from attaining sensitive data. To achieve this requirement, we further divide it into the following sub-requirements:

R1.B.I The need for building DRM mediator by which only it has the capability to render protected contents for a distinct user.
R2.B.II DRM mediator needs to be trusted.

Trust in this context means that the mediator is not subject to change by any illegal entities.

R2 Ensure the authenticity of digital assets, senders and receivers [28].

Authenticating the sender as well as the customers or their devices is a prerequisite requirement for authorizing them. Authentication of the object is required for subsequent use with non-repudiation requirement. This requirement is already achieved by Atwood's architecture [16].

R3 Regulate the legal operation of digital content.

To achieve this requirement, we will use the idea of distributing a license to authorize the use of a content. We further divide this requirement into the following sub-requirements:

R3.A Only legitimate users can gain access to the valid licenses.

R3.B It is recommended that license issuers can account for customers' usages without hurting the scalability of the multicast model.

R4 Ensure the integrity of digital assets [28].
R5 Ensure the non-repudiation for the service [28, 30].
R6 Protect the privacy of end users.
R7 Ensure the availability of the service [28].
R8 Reduce the effect of service compromise.

As we said before, Requirement eight comprises three levels of defence, which are summarized in the following sub-requirements:

R8.A Prevent "break-once, break everywhere".
R8.B Detect and limit the effect of that illegal content distribution.

R8.B is further subdivided into:

R8.B.I End users' Content needs to be distinguished.
R8.B.II This distinction needs to be robust.

R8.C Revise the protection engine once it is being compromized [22, 33, 34].

R9 Support service on demand.

For requirement nine, multicast does not support this requirement any more.

R10 Ensure efficient use of content provider's resources.

For requirement ten, multicast as a technology improves this requirement.

R11 Allow domain access.

This requirement needs to be considered after satisfying the previous requirements (R1...R8).

The next step is a comprehensive study of these requirements.

8 DRM for multicast requirements comprehensive study

From the scalability and content owners point view, multicast distribution gives up the opportunity to improve the performance for some requirements that are met in DRM unicast case, e.g., efficiency requirement (R10). But, it kills some other requirements, e.g., flexibility (requirement R9). In different viewpoint, it can not easily satisfy the BOBE requirement (R8.A), which is solved by individualization in the unicast solution. In the previous section we mapped the requirements for multicast and those requirements that need to be achieved in order to reach the optimal case for DRMM. We will discuss each requirement from the multicast point of view.

R1.B Prevent the action of illegal copying of the clear content from customers' machines.

This requirement is to assure full remote control and "Persistent Access Control" [28] on the digital assets for a limited period of time and is considered the most vital requirement for securing current multicast technologies. Without it, the content owner will not be cheering, if he doubts that his extensive work to produce remarkable product is under control once it comes to the customers' hand. To attain remote control on multicast assets, they must be protected at the network and the application levels.

The most mature scheme to protect multicast content is Secure Multicast [16], but it protects the content at the network level. A sender encrypts the clear content before flowing it into the distribution tree and individually sending these keys to each end user. Because some multicast applications require a dynamic membership, keys may need to be refreshed for every membership change. R1.A is well established by many researchers working in multicast key management and distribution, and we will not go through it.

A legal customer should behave in the way that they should: not copy or redistribute granted contents; unfortunately, bad users do not behave in the way that they should. To keep clear contents away from users' hand and give the rights to legal customers to use them are the means for controlling the use of content. Enforcing these rights would be the responsibility of the network service providers (NSP) and clients' platform. Satisfying both R1.B.I and R1.B.II gives the system the chance to achieve remote control at the application level.

R1.B.I suggests that to remove users' ability to directly access clear contents, it is sufficient to build a DRM mediator to mandate an individual user and give him the capability to legally use the content with considering only legal usages to the service. In this way, the mediator holds up the customer's ability for extracting keys or any sensitive information used to protect the content, e.g., knowing which algorithm is used for content protection, and thus the end user can obtain the clear content with great difficulty only.

OMA DRM 2 specification affects the content protection by keeping the private key inside tamper-resistant hardware, and by hiding the technology secrets used to build the DRM agent. Because the major audience for our proposed multicast content is generic PCs, applying tamper-resistant hardware for multicast content distribution is not feasible because the cost is unaffordable. The major problem with those PCs is that they do not have any special hardware that provides tamper resistance and their operating systems are generic and do not provide any real protection [58].

The feasibility could be achieved when applying tamper-resistant software. R1.B.I proposes to build a DRM mediator to control legal activities. However, a problem arises when a legal customer redirects that mediator to ineligible customers, or extracts the secrets embedded inside the DRM mediator, or tries to modify the logic of the DRM mediator, e.g., modify the part of the code that is resposible of enforcing activities. R1.B.II requires a means to establish a trusted mediator in order to resist the following attacks:

1. Forward the working version of DRM mediator.
2. Reverse engineer the mediator.
3. Modify the logic of the mediator.

Requirement R3 helps the content seller to recover money from the content media; it is to manage the relationship among all parties in the system. We can call it a marketing requirement. In this requirement's view point, the content seller needs to be able to specify the terms and conditions for using the content. The policy server can specify what accounting information needs to be recorded for an accepted End User. This makes it possible, for example, to collect accounting information, to be linked to money. R3 is subdivided into two sub-requirements:

R3.A suggests to use licenses to authorize a legitimate customer to consume the content. The license should describe the legal rights, constraints and include the encryption key. The license should resist being modified or forwarded to unauthorized users [1]. Knowing that licenses are susceptible to analyzing attack, which may allow the attacker to discover and extract encryption keys hidden in the license, we need a mechanism that prevents the customer from tampering with these licenses.

R3.B takes care of another issue, if we adopt the LP role to manage the distribution of the license, we need to deploy it in a way such that the license sender does not need to be aware of the existence of the end user (F1), which contradicts the accounting issue. We need a cost effective mechanism to use licenses as an authorization and accounting mechanism.

R4 is to verify the integrity of the digital assets, security services and the customers' device. Digital assets comprises the content, policy, licence and DRM mediator.

R5 ensures the non-repudiation of requesting a service. A content owner needs its customers to commit to non-repudiation of the request for a service. In the same sense, customers need content owners to commit to non-repudiation of the sending service or any damages could harm customers by using any DRM mediator. This requires that content owners or distributors should not alter the customer's security services.

R6 Protect the privacy of end users. We believe that this requirement has not fully been achieved in the existing DRM model. The content owner needs to trust customers before authorizing them (giving them the license) to use his/her products. In the DRM system, the trust model is based on direct security association between CP, LP and end user.

Therefore, there is one advisor who mandates customers to follow her/his protocol. It is the responsibility of the end user to know who he is dealing with. Multicast deals with more complicated requirements due to the simultaneous multiple users' connection and dynamic membership support. Users may not be aware of the real senders' authority and then they should not be responsible for checking the senders' honesty. Entrusted role could break the whole system. Therefore, trust model has to be changed in a way that follows collaborative protocols between roles and does not reduce the system scalability.

End users need to show their private information to LP as an evidence of their ability to pay for a specific service. Users need be sure that their private information will not be used in a wrong way; this problem is not solved in most DRM systems. Worse than that, users' privacy is under attack, but there is no legal recourse against the attackers [39, 58]. The largest challenge here is that using the DRM mediator makes saving user's privacy harder, because it may hide a rootkit.

R7 Ensure the availability of the service. Here we are talking about preventing DOS attack and maintaining the consistency of CP, LP, digital assets, policies, and licenses. In the DRM system, the services become more available by introducing more servers or caches near to users, as well as by using peer-to-peer distribution, and this will increase the network load, especially when the number of users is inflating. The nature of multicast distibution reduces the extensive interactions between the senders' servers and receivers' machines (F3), which improves the scalability and reduces the number of servers needed to provide such a facility. Multicast introduces two challenges on this issue: (a) the need for maintaining and accounting for end users' behavior requires the interaction between servers and clients; (b) the increasing number of users will increase the probability of DOS attack. This issue was proposed by Islam and Atwood [64].

R8 introduces three lines of defense to mitigate the fact of compromising the service. R8.A is the second line of defense. If the attacker unveils the secrets used to hide digital assets, s/he can affect all the roles (content owners, content distributors and end users), s/he can play the sender role or harm end users and send viruses as well as redistribute content and throw away content owners' money. This tragedy was limited in DRM solution because of the individualization technique. We need to deploy this technique for each individual DRM mediator and license on condition that this individualization should not hurt the scalability requirement.

R8.B is the third line of defense. Content owners spent a long time generating digital contents and if anybody can download them from the Internet for free, then no artist can make any money. Monitoring the Darknet [2] for illegal content distribution and tracing the source of that distribution is the third line of defense and a way to prevent such bad actions. In DRM systems, this could be done by fingerprinting each individual copy. Multicast makes it harder to insert a different mark for each copy, because all customers should receive the same copy. This is one of the big challenging issues.

R8.C is the fourth line of defense, which requires that the system roll back to the previous secure state once it has been compromised. Multicast as a distribution mechanism may provide a promising solution to achieve this requirement.

The previous requirements are needed in the new system, some of them are easy, and others are not.

To conclude, deploying DRM to multicast content needs to distribute the access control to span different administrative points, and needs to provide a consistent access control over digital assets. These distributed administrators need to be concerned with providing the following functionalities: access control management, license management, trust management, system monitor and enforcement management. These functions can be handled by multiple agents in a collaborative process. This distributed management is there in the current DRM systems but the management is housed within a centralized authority (license provider). Distributing the management for multicast session complicates the interactions between interworking parties for controlling multicast content and thus protecting this content becomes harder. Secure multicast sessions need to apply persistant protection on the digital assets in a distributed manner without hurting the multicast scalability requirement.

9 Conclusion and future work

Current scheduled content distribution is not secure and can easily be compromised, a legal customer can redistribute any granted content. We propose the guidelines for using standard DRM means to allow legal customers to do legal activity on the content. Adding DRM to the multicast model will give both models the credibility, and give the CP and content distribution chain the authority to direct the usage for their digital products. By then, they will be sure that the money they spend on developing digital contents will not be thrown away, but it is placed where it will bring the best results.

Satisfying the requirements that are there in DRM model to current multicast model is needed to promote DRM to multicast or vice versa. Multicast improves some of DRM requirements and eradicates others, our future goal is to find a way to mitigate those losses. This mitigation requires the deployment of new technologies that will be feasible, because the revenue outcome will outweight the money needed to deploy these new technologies.

Network service provider should play an essential role and work together with the distribution chain in order to provide the trust for both CP and EU. For future work, we will propose an architecture that satisfies the new requirements and divides the one-to-one trust relationship into cooperative layers that fit the new architecture. At the same time, end users will save money using multicast technology; they just need to be aware of who they are trusting.

Acknowledgements Malek Barhoush acknowledges the support of Concordia University and of Yarmouk University.

Prof. William Atwood acknowledges the support of the Natural Sciences and Engineering Research Council of Canada, through its Discovery Grants program.

References

1. Coyle, K. (2003). The technology of rights: digital rights management. Based on a talk originally given at the Library of Congress. http://www.kcoyle.net/drm_basics.pdf. Accessed 11 Oct 2008.
2. Biddle, P., England, P., Peinado, M., & Willman, B. (2002). The darknet and the future of content distribution. In *Proceedings of 2002 ACM DRM workshop*. http://msl1.mit.edu/ESD10/docs/darknet5.pdf. Accessed 11 Oct 2008.
3. Braden, R. Ed., Zhang, L., Berson, S., Herzog, S., & Jamin, S. (1997). *Resource ReSerVation protocol (RSVP)—version 1 functional specification*. *IETF RFC 2205*.
4. Techtarger (2006). Digital rights management. http://searchcio.techtarget.com/sDefinition/0,,sid182_gci493373,00.html. Accessed 12 Oct 2008.
5. Techterms (2008). DRM (digital rights management). http://www.techterms.com/definition/drm. Accessed 12 Oct 2008.
6. InterTrust. http://www.intertrust.com/main/overview/drm.html.
7. Arnab, A., & Hutchison, A. (2004). Digital rights management—an overview of current challenges and solutions. In H. S. Venter, J. H. P. Eloff, L. Labuschagne, & M. M. Eloff (Eds.), *Proceedings information security South Africa (ISSA)*, Gallagher Estate, Midrand, South Africa. http://pubs.cs.uct.ac.za/archive/00000139/01/arnab_hutchison_issa2004.pdf. Accessed 11 Oct 2008.
8. Huang, T. (2007). Evolution of DRM schema: from encryption to interoperability and monitoring. In *LNCS: Vol. 4577. MCAM 2007*. Berlin: Springer.
9. Deering, S. E. (1989). *Host extensions for IP multicasting*. *IETF RFC 1112*.
10. Multicast Security (MSEC), IETF working group. http://www.ietf.org/html.charters/msec-charter.html. 1 August 2007.
11. Hardjono, T. V. (2004). *The multicast group security architecture. RFC 3740*.
12. Mukherjee, R., & Atwood, J. W. (2007). Scalable solutions for secure group communications. *Computer Networks*, *51*, 12.
13. Islam, S., & Atwood, J. W. (2009). Multicast receiver control by IGMP-AC. *Computer Networks*, *53*(7), 989–1013.
14. Islam, S., & Atwood, J. W. (2008). Sender access and data distribution control for inter-domain multicast groups. *Computer Networks* (submitted).
15. Islam, S., & Atwood, J. W. (2007). A policy framework for multicast group control. In *Proceedings of the IEEE workshop on peer-to-peer multicasting (P2PM2007)*, Las Vegas, NV (pp. 1103–1107).
16. Atwood., J. W. (2007). An architecture for secure multicast. In *Proceedings of the 32nd annual conference on local computer networks* (LCN 2007) (pp. 73–78). doi:10.1109/LCN.2007.123.
17. Hayashi, T., He, H., Satou, H., Ohta, H., & Vaidya, S. (2009). Requirements for multicast AAA coordinated between content provider(s) and network service provider(s). draft-ietf-mboned-maccnt-req (work in progress).
18. Satou, H., Ohta, H., Jacquenet, C., Hayashi, T., & He, H. (2009). AAA and admission control framework for multicasting. draft-ietf-mboned-multiaaa-framework (work in progress).
19. Pankajakshan., B., & Parker., B. J. (2007). U.S. 7,191,332 B1: Digital rights management for multicasting content distribution. Assigned to Sprint Communications Company L.P., Overland Park, Kans. Appl. No. 10/441,748.
20. Hayashi, T., He, H., Satou, H., Ohta, H., & Vaidya, S. (2008). Requirements for multicast AAA coordinated between content provider(s) and network service provider(s). Internet draft draft-ietf-mboned-maccnt-req-06. http://tools.ietf.org/html/draft-ietf-mboned-maccnt-req-06. Accessed 11 Oct 2008.
21. Nickolova, M., & Nickolov, E. (2007). Conceptual model and security requirements for DRM techniques used for e-learning objects protection. *Information Technologies and Knowledge*, *1*(2007), 93–99. http://www.foibg.com/ijitk/ijitk-vol01/ijitk01-1-p18.pdf. Accessed 12 Nov. 2009.
22. Jonker, H. L., & Mauw, S. (2007). Core security requirements of DRM systems. In *Digital rights management, ICFAI book series*, ICFAI, India. http://satoss.uni.lu/members/hugo/publications.php. Accessed 11 Oct 2008.
23. Arnab, A., & Hutchion, A. (2005). Requirement analysis of enterprise DRM systems. In *Proceedings information security South Africa*, Hotel Balalaika, Sandton, Johannesburg. http://pubs.cs.uct.ac.za/archive/00000205/01/arnab_hutchison_issa_2005_final.pdf. Accessed 11 Oct 2008.
24. OMA (2006). OMA DRM requirements. Candidate version 2.1. 10 Oct 2006.
25. Iannella, R. (2001). Digital rights management (DRM) architectures. *D-Lib Magazine*, *7*(6). http://www.dlib.org/dlib/june01/iannella/06iannella.html. Accessed 12 Oct 2008.
26. Sonera Plaza Ltd. (2002). Digital rights management. White paper. http://www.medialab.sonera.fi/workspace/DRMWhitePaper.pdf. Accessed 13 Oct 2008.
27. Liu, Q., Safavi-Naini, R., & Sheppard, N. P. (2003). Digital rights management for content distribution. In *Australasian information security workshop 2003*.
28. Arnab, A., & Hutchison, A. (2007). Persistent access control: a formal model for drm. In *Proceedings of the 2007 ACM workshop on digital rights management* (Alexandria, Virginia, USA). http://pubs.cs.uct.ac.za/archive/00000411/03/acmdrm07-arnab.pdf. Accessed 12 Oct 2008.
29. Lin, E. I., Eskicioglu, A. M., Lagendijk, R. L., & Delp, E. J. (2005). Advances in digital video content protection. In *Proceedings IEEE, special issue on multimedia security for digital rights management*.
30. Onieva, J. A., Lopez, J., Roman, R., Zhou, J., & Gritzalis, S. (2007). Integration of non-repudiation services in mobile DRM scenarios. *Telecommunication Systems*, *35*(3–4), 161–176.
31. Irwin, J. (2004). *The open mobile alliance DRM specifications* (Information Security Technical Report, Vol. 9).
32. Kamat, M. (2002). Security requirement for digital rights management. In *The proceedings of ISECON 2002* (Vol. 19) (San Antonio): 353b.
33. Bar-El, H. Challenges standardizing renewable broadcast security. WhitePaper. http://www.hbarel.com/publications/Challenges_of_Standardizing_Renewable_Broadcast_Security.pdf. Accessed 12 Oct 2008.
34. Grimen, G., Mönch, C., & Midtstraum, R. (2005). Software-based copy protection for temporal media during dissemination and playback. In *LNCS: Vol. 3935. The 8th international conference on information security and cryptology*. Berlin: Springer.
35. Pruneda, A. (2001). Windows media technologies: using windows media rights manager to protect and distribute digital media. MSDN Magazine. December 2001. Microsoft Corporation. http://msdn.Microsoft.com/msdnmag/issues/01/12/DRM/default.aspx.

36. Open Mobile Alliance. http://www.openmobilealliance.org/. Acceseed 12 Nov. 2009.
37. Johansson, J. M., & Grimes, R. (2008). The great debate: security by obscurity. http://technet.Microsoft.com/en-us/magazine/cc510319.aspx. Accessed 27 August 2008.
38. Naumovich, G., & Memo, N. (2003). Preventing privacy, reverse engineering, and tampering. *Computer*, *36*, 64–71.
39. Felten, E. W., & Halderman, J. A. (2006). *Rootkits: digital rights management, spyware, and security*. Los Alamitos: IEEE Comput. Soc.
40. Mori, R., & Kawahara, M. (1990) Superdistribution: the concept and the architecture. *Transaction of the IEICE*, *E73*(7).
41. Arsenova, E. (2002). Technical aspects of digital rights management. http://wob.iai.uni-bonn.de/Wob/images/01212504.pdf. Accessed 13 Oct 2008.
42. OMA (2008). DRM architecture OMA. Candidate version 2.1—05 Aug 2008. http://www.openmobilealliance.org/Technical/release_program/drm_archive.aspx. Accessed 12 Oct 2008.
43. Oma, V. (2008). DRM content format. Approved version 2.0.1.
44. Doehla, S. (2007). DVB-H handheld video content protection with ISMA encryption. http://www.videsignline.com/howto/196801482;jsessionid=NJUVNBYZXPEYQQSNDLPSKHSCJUNN2JVN. Accessed 24 Feb. 2009.
45. Windows Mobile 6 Device Types—Smartphones & PDAs. http://pocketpccentral.net/smartphone/help/general/wm6_naming_scheme.htm. Accessed 31 March 2008.
46. Redmond, W. (2004). Microsoft announces new version of windows media digital rights management software. http://www.Microsoft.com/presspass/press/2004/may04/05-03DigitalRightsManagementTechnologyPR.mspx. Accessed 31 March 2008.
47. Jang, K. W., Park, C. K., Kim, J. J., & Jun, J. J. (2006). A study on DRM system for on/off line key authentication. *IJCSNS International Journal of Computer Science and Network Security*, *6*(2B), 233–238.
48. Arnab, A. (2007). *Towards a general framework for digital rights management (DRM)*. Ph.D., Department of Computer Science, University of Cape Town, 2007. http://pubs.cs.uct.ac.za/archive/00000448/01/alapan.arnab.thesis.final.pdf. Accessed 31 March 2008.
49. Windows Media Digital Rights Management. http://msdn2.Microsoft.com/en-us/windowsmedia/bb190317.aspx. Accessed 31 March 2008.
50. Rosenblatt, B. (2005). Enterprise DRM—technology comparison: authentica active rights management and Microsoft windows rights management services. http://www.drmwatch.com/special/article.php/11579_3519841_5. Accessed 31 March 2008.
51. Microsoft. Windows media rights manager providers. http://www.Microsoft.com/windows/windowsmedia/forpros/drm/supgrade.aspx. Accessed 31 March 2008.
52. Screamer, B. (2001). Microsoft's digital rights management scheme—technical details. http://cryptome.org/ms-drm.htm. Accessed Oct. 27 2007.
53. Ku, B., & Chi, C. (2004). Survey on the technological aspects of digital rights management. In *ISC 2004*.
54. Messerges, T. S., & Dabbish, E. A. (2003). Digital rights management in a 3G mobile phone and beyond. In *Proceedings of the 2003 ACM workshop on digital rights management*, Washington, DC.
55. ISMA (2006). ISMA implementation specification. http://www.isma.tv/. Accessed 23 Feb. 2009.
56. Park, S., Jeong, J., & Kwon, T. (2006). Contents distribution system based on MPEG-4 ISMACryp in IP Set-top box environments. *IEEE Transactions on Consumer Electronics*, *52*(2), 660–668.
57. Housley, R., & Security, V. (2004). *Using advanced encryption standard (AES) counter mode with ipsec encapsulating security payload (ESP)*. *IETF RFC 3686*.
58. Hoglund, G., & Butler, J. (2005). *Rootkits: subverting the Windows kernel*. Reading: Addison-Wesley.
59. Bryant, E. D., Atallah, M. J., & Stytz, M. R. (2004). A survey of Anti-Tamper technologies. CrossTalk. *The Journal of Defense Software Engineering*, *17*(11), 12–16. http://www.arxan.com/ATknowledgePortal/pdfs/Crosstalk-Article-A-Surve-of-AntiTamper-Technologies.pdf. Accessed 13 March 2008.
60. Coyle, K. (2006). Software obfuscation from Crackers' viewpoint. In *Proceedings of the IASTED international conference advances in computer science and technology*, Puerto Vallarta, Mexico.
61. Shapiro, W., & Vingralek, R. (2001). *How to manage persistent state in DRM systems*. (Technical Report). http://citeseer.ist.psu.edu/shapiro01how.html. Accessed 9 Nov 2007.
62. Veeraraghavan, K., Myrick, A., & Flinn, J. (2007). Cobalt: separating content distribution from authorization in distributed file systems. In *FAST'07: proceedings of the 5th USENIX conference on file and storage technologies*.
63. Vadera, A. (1999). SCAMP: Scalable Multicast Protocol for Communication in Large Groups. http://www.cse.iitk.ac.in/research/mtech1997/9711105/all.html. Accessed 12 Oct 2008.
64. Islam, S., & Atwood, J. W. (2006). A framework to add AAA functionalities in IP multicast. In *Proceedings of advanced international conference on telecommunications (AICT'06)*, Guadeloupe, French Caribbean.

Malek Barhoush graduated from Al-Balqa Applied University/Jordan in 1995 with a Bachelor of Computer Engineering and from Yarmouk University/Jordan in 2004 with a Master of Computer Science. He is a Ph.D. Candidate in the Department of Computer Science and Software Engineering at Concordia University, Montreal. His research interests are in digital rights management for multimedia distribution.

J. William Atwood graduated from McGill University in 1963 with a Bachelor of Engineering, from the University of Toronto in 1965 with a Master of Applied Science and from the University of Illinois at Urbana-Champaign in 1970 with a Doctor of Philosophy, all in Electrical Engineering. He is a Distinguished Professor Emeritus in the Department of Computer Science and Software Engineering at Concordia University, Montreal. His research interests include the design, analysis, validation and deployment of protocols for secure multicast, MPLS, and QoS signaling.