트러스트 모델에 기반한 디지털 콘텐츠의 보호

Protection of Digital Contents based on a Trust Model

황성운
ETRI 컨텐츠기술연구부
윤기송
ETRI 컨텐츠기술연구부
김명준
ETRI 컨텐츠기술연구부

Seong-Oun Hwang
Member of Technical Staff, ETRI
Ki-Song Yoon
Principal Member of Technical Staff, ETRI
Myung-Joon Kim
Principal Member of Technical Staff, ETRI

중심어 : DRM, digital contents, trust, security

요 약

인터넷의 광범위한 사용으로 저작권 문제가 대두되고 있다. 본 논문에서는 DRM (디지털 저작권 관리) 기술들과 관련된 주요 이슈들을 기술한다. 현재 시도되고 있는 여러 가지 DRM 아키텍처들을 먼저 살펴보고, 이를 바탕으로 DRM에 적합한 보안 모델 및 아키텍처를 설계하는데 필수적인 보안 이슈들을 논한다. 여기서 도출된 이슈들을 고려하여, 트러스트 모델에 기반한 새로운 DRM 보안 아키텍처를 제안한다.

Abstract

The widespread use of the Internet brings about the issue regarding intellectual property and copyright. This paper describes DRM (Digital Rights Management) technologies and their related issues. Current approaches on DRM architecture are examined first. Based on the review, we discuss some security issues that would be essential to design security model and architecture appropriate for DRM. Having these security issues in mind, we propose a new approach on DRM security architecture, based on our trust model.

I. Introduction

Internet has changed our lives a lot physically and psychologically in a very short time: Internet has made another world of the so-called digital world. It has also made it easier to distribute and exchange information among people. As a result, we can achieve considerable advancement in information technologies. However, behind the bright sides of Internet, there are some problems with Internet that does not exist in real world. One of them is the issue regarding intellectual property and copyright. Digital contents by nature are very vulnerable to unauthorized distribution and use. After a content is downloaded, no further protection is provided on the content that has been accessed. DRM technologies came out to ensure the protection of copyrighted information. Before we go over DRM, It seems good to define DRM. Here we cite a definition on DRM. DRM (Digital Rights Management) is management of the creation, manipulation, distribution, and consumption of digital information. It enables exchange of value for use of digital information such as payment, usage data[NG]. In other words, DRM can be defined as a set of technologies that collectively support all the life cycle of content - creation, manipulation, distribution, and consumption - by doing prevention of illegal copying, imposition of fee, and processing of payment as well as protecting each principals right and profit. Principals may include any participants involved in the contents life cycle, such as content creators, providers, distributors, users, right holders, etc. DRM products basically provide persistent protection on DRM-protected content. The users behaviors on contents such as buying, copying, printing, redistributing, the length of time available, the permissible or remaining count of use, limitations of the device used are checked and enforced by DRM products according to the contents associated usage rules. DRM products, often through contact with a financial clearing house, oversees the payment process so that everyone involved in the creation, production, and distribution of the content is paid fairly. In addition, DRM products can allow tracking and reporting of consumer preferences and buying patterns, providing what could be a valuable source of marketing information. However, the current DRM technologies are not complete in the aspect of right protection. It is known that hackers have already reverse-engineered the DRMprotecting code on DVDs. In general, DRM protection technologies are based on cryptology[JBK00].

Our final target is to design a new security model and architecture that are appropriate for DRM and to implement it. As a preliminary step, this paper examines DRM-related security issues and DRM architecture. This paper has the following structure: In section 2, we examine current approaches on DRM architecture. Section 3 discusses some security issues that should be considered when designing a DRM system. Section 4 proposes a new approach on DRM architecture, based on our trust model. Section 5 concludes this paper with some discussion.

II. Current Approaches on DRM System Architecture

There are largely two kinds of DRM system, server-based DRM and client-based DRM. Sever-based DRM architecture is taken by most DRM solution providers such as InterTrust, Microsoft, Contentguard, and IBM. Client-based approach is supported only by InterTrust.

In the server-based approach (refer to Fig. 1), the so-called license server manages users licenses and their related information. In this model, users request licenses to server to play some media content, in return paying for the license. Then license server issues to users licenses that describe usage rule associated with the content.

Every operation is intercepted by an Access Control Enforcement Facility (AEF), which asks an Access Control Decision Facility (ADF) for a decision about whether this action is authorized. While users are playing the content, the usage information is sent to the license server. In this way, users usage information are kept and managed at the license server. The strength of this approach is simple and easy to design and to implement, while the weakness is that users should always be connected to the server whenever they play the content.

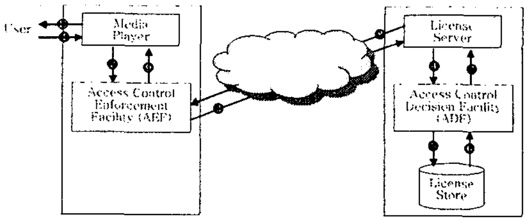
User
Media Player
Access Control Enforcement Facility (AEF)
License Server
Access Control Decision Facility (ADF)
License Store

Fig 1. Server-based DRM Architecture

Note that all the figures shown in this paper are results produced while we configure activities between DRM client and server under the ISO Access Control Framework[ISO92] model. The terms AEF and ADF come from it. The attached numbers on the figure show the order in which access control is performed.

In the client-based approach (refer to Fig. 2), license management is made in the users environment.

Briefly explaining, except the first time when users get licenses from license server, usage information and license management are made locally through specific secure module such as AEF, ADF, and Secure DB that are located on the users devices. This makes it unnecessary for users to always connect to the license server. In the users aspect, this approach is more preferable than server-based model.

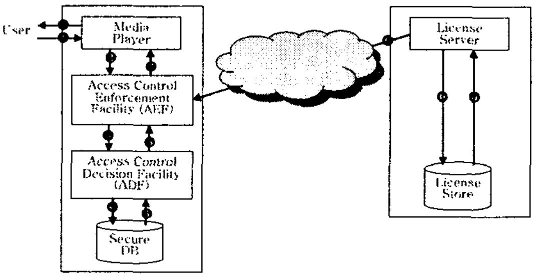
User
Media Player
Access Control Enforcement Facility (AEF)
Access Control Decision Facility (ADF)
Secure DB
License Server
License Store

Fig 2. Client-based DRM Architecture

However, since important information like license, usage count are kept in client-side, the possibility of attack is higher. Implementation is not so easy compared to server-based model.

In the client-based DRM model, users or hackers attacks against AEF, ADF, and Secure DB are expected to have the form of reverse-engineering attack or tampering attack. Researches are being done on these issues. (Refer to www.cloakware.com<http://www.cloakware.com>, www.intertrust.com <http://www.intertrust.com>, respectively.)

III. Security Issues in DRM Systems

1. Paradigm shift of access control mechanism

Reference Monitor that consists of both AEF and ADF has been located in server-side in most client-server environment. However, in the case of client-based DRM, both or considerable portions of the above functions that Reference Monitor has are moved to client-side. As a result, the possibility or severity of attack by clients or hackers is higher than before. To make it worse, the existing Reference Monitors are basically based or supported on the OS (Operating System) level, while one in client-based DRM is based on application of DRM client, e.g., InterTrusts DigiBoxTM. The key questions in building DRM system are What rules must the system establish? and How can we assure that the system enforces them? Although establishing rules by itself is important, it seems that more emphasis should be on how to ensure of enforcing the rules compared to the previous access control mechanism.

2. From server-centered to user-centered security mechanism

Most of security systems have been designed and implemented server-oriented or server-centered. In other words, users register and are granted (or given) necessary access rights by system administrator or system security officer. The situation becomes somewhat different in DRM system. In DRM system, unlike the previous existing service, the service is exchanged for payment by user or usage data. Therefore, when we design security model for DRM system, if we dont take into account user-centered design of security, no one will not choose the service. User-centered design will help to improve the understanding of the system, at least having the following features: It should provide protection control mechanism of users privacy against information leakage or misuse; The process of registering and getting licenses should be easy enough to users; The security model should be understandable to users; One more thing, until now users have had no choice but to accept rules and services pre-determined by server. Now it should enable users to establish rules and services by negotiating with server.

3. Disciplined entity authentication/authorization required

Most of previous authentication mechanisms can be said static, close model. In those models, subjects, objects, users, etc. are determined in advance and their roles or access rights are very restrictive or non-flexible. However, those models are difficult to be applied in DRM system model. It seems that it needs some kind of modification. In the DRM system environment, there are various entities such as, for example, according to IMPRIMATUR business model[IMP99], content creator, provider, distributor, purchaser, CA (Certificate Authority), Clearing House for managing finance and usage right, Right Holder, IPR (Intellectual Property Registration), Monitoring Service Provider, Unique Number Issuer. There could be more entities involved depending on specific business model. Furthermore, there could be various kinds of rights between those entities. Therefore new entity authentication /authorization mechanisms are required to support the value chain. In particular, it should be possible that several entities could negotiate and cooperate to produce business rules to enforce transparently on some digital content. New mechanisms will be basically of the form of distributive, cooperative computing model. New mechanisms should be device-independent. That is, they should support diverse devices of entities, e.g. they could support transparently diverse users devices such as PC, PDA, mobile phone, DVD, MP3 player, etc. In the disciplined way, they should support both entity authentication and right authorization.

4. Previous, existing security models are not good enough for DRM

Until recently, researches on security model have been largely limited to models of confidentiality[Lam71][GD72][BLP75][HRU76][GM82][McL90]. In other words, access control models for enforcing confidentiality have dominated in the direction of computer security research. Everyone agrees that access control model have contributed to computer security area by establishing both theoretical and practical structure. Those initial security researches had been originated in the military or by US national funding that is not so much different from the military. However, if we apply some of those research results in the commercial world, we face the following issues. Previous security model structures are very static in the sense that security rules are established to keep confidentiality or integrity well in advance, and users or participants have only to follow the rules. Thats all. Maybe thats enough at least in the military. When we discuss desirable properties of security model, we basically discuss generality, predictive ability, and appropriateness (the estimation of how well it describes the application systems activity structure and features). However, in the commercial world, a security model should take users into account. Here, a user could be a person or an organization or a set of organizations. In other words, the security model should be meaningful and more understandable to users. Most of previous security models describe the relationship between system and users (so-called closed, non-flexible relationship structure), while we need to describe additional relationships like relationships between users, relationships between systems, (so-called open, flexible relationship structure) and so on.

5. DRM system needs some trust model

Related to this, we can see some examples, e.g., we often see book reviews when we go to online book stores or we see searched result with matching probability after performing some search engine. As in the same way, to describe digital world well in DRM system, we need a systemic, trustworthy trust mechanism that enable entities to evaluate the trust level of corresponding entities, and the quality of some digital content, to consult, collect and aggregate (or categorize or classify), and to provide (or transfer) the trust information to other requesting entity.

IV. A New Approach on DRM System Architecture

In this section, we propose a new approach on DRM system architecure, based on trust management. Fig. 3 briefly shows our proposed approach based on trust management, which is applied in the client-based DRM architecture. Note that our approach can also be applicable to the server-based DRM architecture with some modifications.

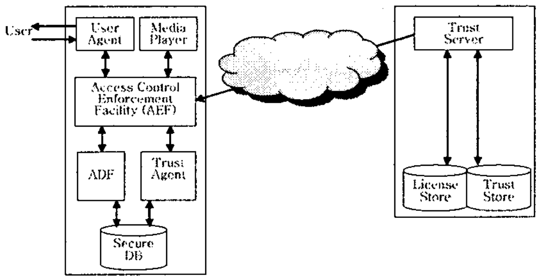

Fig 3. Proposed DRM Architecture

User Agent and Trust Agent are added on client-side. On server-side, the previous License Server is replaced with Trust Server and Trust Store is added. The existing approach, regardless of whether it is client-based or server-based DRM, is based on license. Users pay for the media content and in return they receive licenses that enable them to play the media content. Licenses are usually represented (or implemented) by a decryption key to decipher the encrypted content. However, under our proposed model, users can use their trust as well as their licenses as in the existing approach. At the first time, users pay for the selected content and get licenses as the client-based approach way. Users can enjoy the content until their licenses expire or the usage counts exceed over the pre-defined values. In addition, our approach could permit their continuing use if their trust levels permit. That is, while users spend their licenses, their trust value are calculated and saved at their Secure DB. Trust value can be calculated between entities according to some defined *trust model, which will be described later.*

In the following, we briefly explain some major functions of modules depicted in Fig 3. User Agent helps users to set up their own personalized environment. Through User Agent, users can designate their own payment method, e.g., bank name, account number or credit card number. Users can also set up their preferences through negotiating with other entities. Preferences could designate what kind of personal information could be provided with the content provider for the protection of the users privacy as well as for getting the better, personalized service based on users profile information. Only users can access their own User Agent. Any kind of access to User Agent is not allowed for the other entities. Another important role of User Agent is to monitor the communication path between AEF and Trust Server in order to detect possible leakage of users privacy information. Trust Agent has a role of calculating the users trust value and save it at Secure DB, which can be accessible by Trust Server. Sometimes Trust Agent communicates with Trust Server and updates or reflects their trust values according to the received trust value from Trust Server. Secure DB keeps users (or owners) trust values as well as license information together with usage information. Trust Server has both roles of license management and trust management. On the client-side, only User Agent exists on behalf of users. The other components like AEF, ADF, Trust Agent, and Secure DB exist for servers interest and should be robust against attacks from users or hackers.

1. Proposed Trust Model

A trust relationship exists between entity A *and* B *when* A holds a belief about Bs trustworthiness. It is said that A is a trusting entity (subject) and B is a trusted entity (object). Entities use trust domains to express which aspects of trust they are referring. Trust values or levels are used to express the different values or levels of trust an entity may have in another. For example, we trust a CA (Certificate Authority) to certify public keys (domain Sign key), while a specific content provider can assign trust value to its purchaser (or user) when some conditions are met, e.g., the *user obeys the usage rule embedded in the content or* recommends their content to other users in a form of super-distribution that means content is delivered from person to person in a trusted and accountable manner. Super-distribution may accelerate the distribution of content for purchase. Based on users trust level, content providers can provide users with distinguished, personalized service, e.g., users with high trust level can get a discount when purchasing another content. On the contrary, some bad users that did not pay for his license or make illegal copy or infringe copyright could get low trust value. Table 1 shows the elements of the proposed trust model.

Table1. Elements of the Model

Set	Elements	Semantics
S	$\{S_1, S_2, \ldots, S_n\}$	Subjects: entities who evaluate other entity's trust values, that is, evaluators.
O	$\{O_1, O_2, \ldots, O_m\}$	Objects: entities who get trust evaluations from other evaluators.
D	$\{D_1, D_2, \ldots, D_k\}$	Domains: evaluation areas under which trust evaluations are made.
C	$\{C_1, C_2, \ldots, C_l\}$	Criteria: evaluation check items on which trust evaluations are made. For simplicity, we assume that each domain has only one set of criteria.
I	$\{I_1, I_2, \ldots, I_l\}$	Importance Factor: weighted values (real numbers ranging between 0 and 1) assigned to each check items of Criteria. These values usually are determined by each evaluator, so they could be different from evaluator to evaluator.
T	$\{0, \ldots, t, \ldots, 1\}$, $0 \le t \le 1$	Trust values: assigned to objects by evaluators. $t \in T \subseteq S \times O \times D \times C \times I \times [0, 1]$

2. Evaluating Trust

Trust values can be computed in the following cases:

CASE 1: When an entity comes into a new domain without any prior trust history, he/she needs to compute trust values on the existing entities of the domain and the other entities also need to compute a trust value on him/her.

CASE 2: When an entity in the domain continuously update his/her trust values on other entities in the same domain, In CASE 1, when an entity j comes into a new domain, he can compute trust values on the existing entities using the available trust values on the domain as in the following: ti,j, a trust value assigned to j by an evaluator i,

$$t_{i,j} = \sum_{k} [\{(\sum_{r} t_{r,k})/|N_k|\} \bullet t_{k,j}] / \sum_{k} [(\sum_{r} t_{r,k})/r].$$

$$k \in N_j, r \in N_k \text{ ....}(1)$$

where Nj, the set of entities which evaluate trust values on an entity j, and |Nj|represents the number of elements of the set Nj. In the other hand, for simplicity, we assume that other entities take default trust value (e.g., 0.5) for the newcomer j.

In CASE 2, an entity could update his trust value tnew considering his previous one told and his recently computed (expected) one texp or recommended one trec from other entities using the following exponential average:

$$t_{new} = \alpha \bullet t_{exp} + (1 - \alpha) \bullet t_{old}$$

$$\text{or } t_{new} = \alpha \bullet t_{rec} + (1 - \alpha) \bullet t_{old} \text{ ....}(2)$$

, where the parameter α, $0 \leq \alpha \leq 1$, controls the relative weight of recent(or recommended) and past history in our prediction.

Here is a sample example to show the trust evaluation procedure. The following tables show the trust evaluation relations of the entity set {e1, e2, e3, e4} in some domain.

Table 2. Trust Relationship

	e_3	e_4
e_1	0.7	0.8
e_2	0.9	0.6

	e_1	e_2
e_3	0.5	0.4
e_4	0.9	1.0

In the table 2, the entities on the column and row represent evaluators and the targets of evaluation, respectively. For example, the entity e1 assigns a trust value of 0.7 to e3.

When a new entity e5 computes trust value t5,3 on e3, it follows equation (1):

t5,3 = [{(0.5+0.9)/2}0.7+{(0.4+1.0)/2}0.9]
/ [(0.5+0.9+0.4+1.0)/2]
= 0.8

Next time, when entity e5 updates his trust value on e3, based on a recommenders trust value 0.7 and assuming α = 0.8, it follows equation (2):

t5,3 = 0.8 0.7 + (1-0.8) 0.8 = 0.72.

Note that α is usually determined freely by evaluators, but in case of trust value computation based on recommended one, it can be computed from equation (1).

When entity e5 updates his trust value on e3, based on the current trust information described in the table 3, it follows equation (2). The following table shows the collection of measurement data per each criteria item and its revised results according to importance factor for evaluator e5 to evaluate e3.

Table 3. Criteria - Measurement

criteria	Importance factor	Measured data	Revised results
c1	1.0	0.7	0.7
c2	0.8	0.9	0.72
c3	0.9	0.8	0.72

Revised results are computed by multiplying importance factor and measurement data together to reflect the importance of each item in the criteria set. So we can compute texp = (0.7+0.72+0.72) / 3 = 0.713. When assuming told = 0.72, α = 0.8 as computed earlier,

$$\begin{aligned} T5,3 &= \alpha \cdot texp + (1 - \alpha) \cdot told \\ &= 0.8\ 0.713 + (1 - 0.8)\ 0.72 \\ &= 0.714. \end{aligned}$$

VI. Discussion and Conclusion

In the paper, we discussed security issues to be considered when designing a DRM system. We also examined current approaches on DRM architecture and proposed a new approach based on trust management. Our imminent work would be to implement the concept of our trust model in DRM system. The trust model will serve as an infrastructure of digital content distribution network or community. In other words, the trust model will support active, transparent distribution of digital contents as well as protecting the digital copyrights of DRM-protected contents.

[NG] Neal Glew. InteTrust, TDB and Object Encodings. InterTrust Technologies Corp. Refer to http://www.intertrust.com.

[JBK00] Joonsang Baek, Byoungcheon Lee, and Kwangjo Kim, Provably Secure Length-Saving Public-Key Encryption Scheme under the Computational Diffie-Hellman Assumption, ETRI Journal, Volume 22, Number 4, December 2000.

[ISO92] *International Organization for Standardization. Information Retrieval, Transfer and Management for OSI: Access Control Framework, ISO/IEC JTC 1/SC 21/WG 1 N6947 Second CD 10181-3, May 1992.*

[IMP99] IMPRIMATUR. The Business Model Synthesis, January 1999. Refer to http://www.imprimatur.net.

[Lam71] B.Lampson. Protection. In 5th Princeton Symposium on Information Sciences and Systems, March 1971. Reprinted in ACM Operating Systems Review, 8(1), 1974.

[GD72] G.Graham and P.Denning. Protection principles and practice. In Proc. Spring Joint Computer Conference. AFIPS Press, 1972.

[BLP75] D.Bell and L.LaPadula. Secure Compupter System: Volume II. MITRE Technical Report 2547. Reprinted in Journal of Computer Security, 4(2/3), pages 239-263, 1996.

[HRU76] M.Harrison, W.Ruzzo, and J.Ullman. Protection in operating systems. Communications of the ACM, 19(8):461-471, August 1976.

[GM82] J.Goguen and J.Meseguer. Security policies and security models. In Proceedings of the 1982 IEEE Symposium on Research in Security and Privacy. IEEE Computer Society Press, 1982.

[McL90] J.McLean. The specification and modeling of computer security. Computer, 23(1):9-16, January 1990.

황 성 운 (Seong-Oun Hwang) 정회원

1993년 8월 서울대학교 수학과(학사)
1998년 2월 포항공과대학교 정보통신학과(공학석사)
1994년 1월 ~ 1996년 2월 LG-EDS Systems, Inc.
1998년 1월 ~ 현재 한국전자통신연구원 연구원
<관심분야> : 암호 이론, 보안, DRM

윤 기 송 (Ki-Song Yoon) 정회원

1984년 2월 부산대학교 조선공학과(학사)
1988년 2월 City University of New York (전산학 석사)
1993년 2월 City University of New York (전산학 박사)
1993년 3월 ~ 현재 한국전자통신연구원 책임연구원
<관심분야> : 정보보호, 메시징, 분산처리

김 명 준 (Myung-Joon Kim) 종신회원

1978년 2월 서울대학교 계산통계학과 졸업 (학사)
1980년 2월 한국과학기술원 전산학과 졸업 (이학석사)
1986년 5월 프랑스 Nancy 제1대학교 응용수학 및 전산학과 (이학박사)
1980년 2월 ~ 1981년 6월 아주대학교 종합연구소 연구원
1981년 10월 ~ 1986년 5월 프랑스 Nancy 전산학연구소 (CRIN) 연구원
1986년 7월 ~ 현재 한국전자통신연구원 책임연구원
1993년 프랑스 Nice Sopia-Antipolis 대학 초빙교수
<관심분야> : 데이터베이스, 소프트웨어공학