Privacy Preserving Multiparty Multilevel DRM Architecture

Amit $Sachan^a$, Sabu $Emmanuel^a$, Amitabha Das^a, Mohan S $Kankanhalli^b$
[a]School of Computer Engineering, Nanyang Technological University, Singapore
[b]School of Computing, National University of Singapore, Singapore
{amit0009, asemmanuel, asadas}@ntu.edu.sg, mohan@comp.nus.edu.sg

***Abstract*— Traditional digital rights management (DRM) systems are only two party systems, involving the owner and consumers. However, for scalability of business it is often necessary to involve additional levels of distributors and sub-distributors, who can promote and distribute the content in regions unknown to the owner. Thus, we propose an architecture for multiparty multilevel DRM system. The term 'multiparty' refers to involvement of many parties such as the owner, distributors, sub-distributors and consumers and the term 'multilevel' refers to multiple levels of distributors/sub-distributors. The architecture also supports the log files based violation detection, in case of violation of DRM system by any party. However, violation detection imposes a problem of preserving privacy of consumers. So, in the architecture, a provision is made to preserve their privacy.**

I. Introduction

Great improvement in network technologies has made the Internet a convenient and cheap mode of multimedia content distribution. However, it has also increased the fear of illegal copying and redistribution. Digital rights management (DRM) is the technology that intends to restrict the illegal media consumption, copying and redistribution by the use of digital licenses. Various DRM architectures have been proposed for digital content and license distribution for a typical two party scenario, where the owner and consumers are the only parties involved in the architecture [1] [2] [3] [7] [8]. However a two party architecture may not be sufficient to provide proper business scalability as it is too restrictive and often it is not able to make proper business strategies for all regions and cultures. This is the reason for almost 90 percent of sale of digital music still being concentrated only in top 10 markets [12]. To allow for more innovative and scalable business models which have the flexibility of packaging multiple contents together in a regional and culturally sensitive manner it is necessary to have a more flexible and hierarchical distribution network. Hence, a multiparty, multilevel architecture involving multiple levels of distributors and sub-distributors (lower level distributors) in addition to the owner and consumers is necessary. A local distributor can better explore potentially unknown market to the owner and make strategies according to the market. In addition distributors can also help in handling different price structure of media in different countries, and in case of price or demand fluctuation cost may be shared between the owner and distributors. In this paper, we propose a multiparty, multilevel DRM architecture. Though the architecture helps the owner in establishing business better, it presents new requirements such as: to find a content packaging mechanism suitable for all parties; trust management between all parties in the architecture; to preserve privacy and rights of each party involved in the architecture. The objective of our proposed architecture is to take care of all these requirements. The rest of this paper is organized as follows: section II presents requirements for multiparty, multilevel DRM architecture. Section III discusses proposed DRM architecture. In section IV, we describe violation detection in the architecture. Privacy issues are discussed in section V. In section VI, we discuss related works. Finally, the paper is concluded in section VII.

II. Requirements of the Architecture

The parties involved in a typical multiparty architecture[9] are the owner, distributors, and consumers. Each of the party has its own concerns and requirements that need to be satisfied by the architecture. Below, we describe concerns and requirements of each party in such an architecture.

The Owner: The owner is concerned about unauthorized usage (such as play, copy, etc, without having permissions to perform) and illegal redistribution of contents. To resolve the owner's concern about unauthorized use of content, firstly, it is required that content must be encrypted with the owner's key. Secondly, digital licenses are required that contain different permission and constraints associated with the permissions. Thirdly, a trusted DRM agent is required at each party that can perform actions on contents according to the licenses. Finally, if there is some unauthorized use due to system violation then it should be detected. We assume that unauthorized use can be detected with the help of usage logs as usage logs will reflect actual activities of consumers. So, usage logs should be created at consumers' machine. To collect and analyze the logs an entity called log collection center should be established by the owner. To resolve the owner's concern over illegal redistribution, the owner's content encryption keys should not be disclosed to any distributor or consumer. For this purpose, the owner can employ a trusted third party (TTP), called license server. It should store and serve content decryption key directly to consumers when instructed by the owner or distributors. Trusted DRM agent at consumers'machine also prevents disclosure of keys to consumers.

Distributors: Each Distributor in the architecture maintains their own content server (CS) separate from that of the owner. So, they are concerned about proving their distributorship to

other parties, and protection of contents in their CS from being downloaded by consumers of other distributors. This concern is because a selfish distributor can redirect his consumers to other distributors' CS for downloading contents without sharing profit with them. To resolve distributors' concerns, a content packaging mechanism is required such that the contents in distributors' CS cannot be used by the owner and other distributors, and their distributorship can also be proved.

Consumers: Consumers are concerned about ease of getting contents, ease of usage of contents and their own privacy. To resolve concern on ease of getting contents, it is required that the architecture should support the super-distribution technique in addition to direct download from a CS. The super-distribution technique allows consumers to exchange the contents among themselves easily. Requirements for ease of content usage are discussed in the section III B.

In addition, all the parties in the architecture have a concern about authentication of other party during any communication. For authentication purpose, we can make use of any existing PKI infrastructure (such as X.509, or PGP)[10]. A TTP, called certification authority (CA), in the PKI infrastructure can issue digital certificates for authentication purpose.

III. Proposed DRM Architecture

Our proposed multiparty, multilevel DRM architecture consists of a distribution chain part and an administrative part as shown in figure 1. A distribution chain is the chain of intermediate parties, where each party passes content to a party next to it before the content reaches the consumer. Depending upon the length of a distribution chain, parties involved may be the owner O, multiple levels of distributors D_{ij} and consumer C; where D_{ij} represents j^{th} distributor at the i^{th} level. Number of levels of distributors in a distribution chain may depend upon the extent of the region to be explored and density of consumers. The owner and distributors also maintain their own CS. The CS are represented by CS0 for the owner and CS_{ij} for the distributor D_{ij}.

The administration part consists of registration authority (RA), license server (LS) and log collection center (LCC). The RA does registration of parties involved in the architecture. For registration purpose a party generates its public/private key pair and sends the public key along with its identity, and device ID to the RA. Next, the RA sends digital certificate request to the CA. The CA then generates and signs the certificate and sends it back to the RA. The certificate is then served by the RA to the party concerned. Digital certificates are used for authentication purpose in the architecture. The LS signs and serves the licenses created by the owner/distributors to license requesting party. The LS also keeps record of permissions and constraints involved in each license. The LCC collects logs from consumers and LS, and does processing of the logs collected to make further business strategies and to detect violation of permissions and constraints by any party in the architecture. Next, in this section, we discuss in detail various functions, such as content distribution, content packaging, licensing, and content consumption.

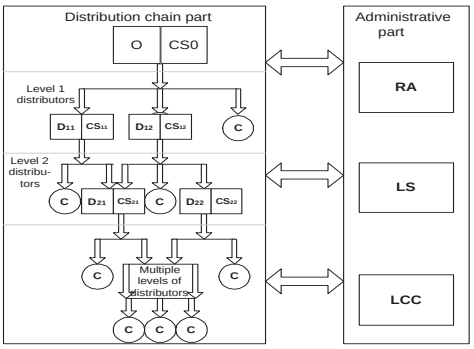

Fig. 1. Multiparty Multilevel DRM architecture

A. Content Distribution

The owner is primarily responsible for distribution of content. The owner can appoint various distributors to facilitate the distribution process, and distributors are also allowed to appoint next level of distributors. The owner and distributors can distribute contents through their own CS and they can also rely on super-distribution technique. Consumers can download content from their preferred distributor/ owner's content server or get it through super-distribution.

B. Content Packaging

The proposed architecture supports the packaging of different types of media contents such as video, audio, text and image files. A typical content package P consists of the content *K* and associated header H_K. Thus, the content package can be represented as: $H_K \| K$. Content header H_K contains metadata about the content *K*, information to play the content, information about the compression algorithm, etc.

The second part of content packaging is the encryption of the content. The content encryption mechanism must be such that it resolve the owner's concern over security of content and distributors' concerns over illegal download of content from their CS, as discussed in section II. To resolve their concerns, we propose content encryption scheme based on two content encryption keys. The first one, called *global encryption key* (GEK), is unique for a particular content and the owner generates this key. The second key, called *local encryption key* (LEK), is different for the owner and different distributors and different contents. The objective of the GEK is to prevent unauthorized use of contents, and the objective of the LEK is to prevent overloading of content servers of the owner and distributors by consumers of other distributors.

As the first step of the content encryption the owner encrypts content K_m with GEK_m (GEK for the content K_m) without encrypting the format header H_k of the file. Thus, after the first step, format of the file is preserved. Symbolically, the content K_m after encryption can be shown as:$H_{K_m} \| E_{GEK_m}[K_m]$

In the next step the owner encrypts initially encrypted content along with format header again with LEK_{o-m} (i.e. LEK of the owner for the content K_m). And then a new header, called *package header*, H_{o-m} is appended to the content so that DRM agent can recognize it as a valid content encrypted

with GEK and LEK. The package header H_{o-m} also provides information about content download URL, license acquisition URL and other metadata related to the owner. Thus obtaining the content as: $H_{o-m} \| E_{LEK_{o-m}}[H_{K_m} \| E_{GEK_m}[K_m]]$.

When a distributor purchases distribution rights for content from the owner or other distributor, the distributor is allowed to decrypt the content using LEK of the owner/distributor concerned. The distributor gets content only encrypted with GEK and follows process similar to the process followed by the owner in the second step of content encryption before uploading the content into his content server. Thus, the content at j^{th} distributor in the i^{th} level can be represented as: $H_{D_{ij}-m} \| LEK_{D_{ij}-m}[H_{K_m} \| E_{GEK_m}[K_m]]$. Where $H_{D_{ij}-m}$ is the new package header, and $LEK_{D_{ij}-m}$ is the LEK of the j^{th} distributor in the i^{th} level for the content K_m.

Distributors are also allowed to combine two or more contents together to make a package. Packaging multiple contents provides additional flexibility to distributors while making strategies for a particular region as distributors can package several contents in regional and culturally sensitive manner. A typical package of two contents K_m and K_n can be represented as: $H_{D_{ij}-(m,n)} \| LEK_{D_{ij}-(m,n)}[H_{K_m} \| E_{GEK_m}[K_m] \| H_{K_n} \| E_{GEK_n}[K_n]]$ Where $LEK_{D_{ij}-(m,n)}$ is the LEK of the j^{th} distributor in the i^{th} level for the package.

The scheme presented above also resolves distributors' concern about proving their distributorship as content cannot be used without the GEK, which can only be obtained from the LS. Involvement of the LS provide assurance about legal distributorship to other parties. However, double encryption in the scheme presented above can raise consumers' concern over ease of usage of contents. But, we can make use of the fact that aim of the LEK is only to protect CS misuse. So, we can use a lightweight selective encryption algorithm such that it can distort viewing experience of consumers redirected by other distributors. For example, encrypting 10 seconds of I-frames in every 1 minute video may be sufficient to distort the viewing experience of consumers. Assuming 30 frames per second and size of GOP(group of pictures) as 12. The selective encryption is needed for 25 frames out of 1800 frames per minute. It will increase complexity only by 1.38 percent as compared to single encryption for all the frames. The proposed solution does not affect security of the content as security will be at least that provided by the GEK.

C. Digital Licensing

License is a digital document used to establish an agreement between two parties. In the agreement the license issuer allows the other party (license requesting party) to access the content. Content access is regulated with the help of permissions, constraints and content decryption keys. Permissions correspond to actions that can be performed on the content e.g. play, copy, edit, reuse, and redistribute. Constraints are limitations associated with the permissions in the license. Various types of constraints can be time based, count based, location based, etc. Content decryption keys are used to decrypt encrypted content and are available for a particular permission only if all the constraints associated with that permission are satisfied. For example if a license is issued with play permission with constraints of 10 count and validity of 30 days. The decryption keys in the license will be unavailable for play permission after 30 days even if less than 10 counts are used.

1) License Creation: License is created by the owner and distributors for other distributors and consumers. License contains the following entries: Name of license issuing party, ID of content(s), Permissions, Constraints, and Keys required for taking appropriate action. In this architecture two types of licenses can be created by the owner/distributors. The first type of license is redistribution license and the other is usage license. Both the types of licenses are described below:

(i) Redistribution licenses: Redistribution licenses are created by the owner or a distributor for another distributor lower in the distribution chain. The redistribution licenses contain LEK of the party which generates license for a particular content package. Typical permissions in this type of license are permission of content redistribution and permission to issue redistribution licenses. Content redistribution permissions enable distributors to create their own content package from one or more contents and upload the content on their own content server. Permission to issue redistribution license allows distributors to accommodate additional levels of distributors. Typical constraints associated with permissions in this type of licenses are time based, count based, and location based. Enforcement of redistribution license is done with the help of license server, which keeps record of redistribution and usage licenses issued by owner/distributor.

(ii) Usage licenses: Usage licenses are created jointly by the owner/distributor concerned and the LS. Involvement of the LS in license creation is necessary as distributors don't know the GEK. Usage licenses contain both the GEK and LEK of the owner/distributor for a particular content. With the help of usage license consumers can use the content according to permissions and constraints in the license. Typical permissions in usage licenses are play, copy, print etc. Enforcement of usage license is done with the help of trusted DRM agent at consumer's machine.

2) License Distribution: The LS does the distribution of both types of licenses. And it can be done in two ways. The first way is *readymade license* distribution and the second way is *on-demand license* distribution. In the readymade license distribution method certain predefined licenses for all the contents are already created and stored at the LS. A distributor/consumer selects preferred type of license on the owner/distributor's website. The owner/distributor then redirects distributors/consumers to the LS. In the on-demand license distribution, the owner/distributor creates a license for each license request made by a distributor/consumer. For this type of license distributor/ consumer can negotiate for permissions and constraints with the owner/distributor. The owner/distributor creates the license with requested permissions. After creating the license, the license generating party sends it to the LS and redirects requesting party to the LS. In

both the methods licenses are served by the LS to requesting party through a secure channel. Both the distribution methods have their own advantages. The readymade license distribution method has less computational and communication complexity since the owner/distributors do not need to create a license for each license request. Whereas, the on-demand license distribution method provides flexibility to distributors/consumers in choosing various permissions and constraints.

D. Content Consumption

Content consumption by consumers is a two step process. The first step is license acquisition and the second step is license execution to use the content. Next, in this section we describe both these steps in detail.

1) License Acquisition: Acquisition of license is required to use the content. In order to acquire license consumer needs to connect to a higher party (owner/distributor) in the distribution chain using license acquisition URL, which is in the header of the content. After getting license acquisition request from the consumer the higher party first checks registration status of the user and if the consumer is not registered then the consumer is redirected to the RA to get registered. If the consumer is already registered then the higher party redirects the consumer to the LS for acquiring the license.

2) License Execution: A trusted DRM agent at the consumer's device is required for usage of contents. The DRM agent first decrypts the license using its private key. After that the DRM agent verifies the signature of the license server on the license. If the signature is verified correctly then DRM agent adds the license in the pool of licenses, which is kept in a secure database. When the consumer needs to play the content, the DRM agent decrypts the content using the LEK and GEK in the license. The DRM agent also maintains a registry, which contains remaining permissions of active licenses. When the consumer uses a media, the DRM agent reads the license, evaluates rights and adjusts the remaining permissions and constraints associated with permissions appropriately. When any of the constraints is not satisfied, the DRM agent must not consume the content. When a license expires, the DRM agent removes the license from the registry.

IV. DRM System Violation Detection

DRM systems are vulnerable to attacks due to the programming, operating system, or hardware based loopholes [5]. For example, system date change in some older DRM systems may allow consumers to use the contents for longer period, or, modification in system registry may allow consumers to bypass count restriction. In a multiparty DRM system distributors can also violate the system by not restricting themselves to the permissions and constraints in their redistribution license. For instance, suppose a redistribution license allows the distributor to sell 1000 copies of content in Asia within 1 month. Now, if the distributor sells more than 1000 copies, or sells any copy outside Asia, or sells a copy after 1 month using the license then it would be a violation. Below, we present a framework for detection of violation by consumers and distributors.

Log files created at the consumers' machine and LS can be useful to detect violation. The log files at consumers' machine contain data related to usage history. These log files are automatically created or appended by the DRM agent when certain actions such as play, copy, etc are carried out on the content. The log files at the LS contain permission and constraints granted by the owner/distributors to other distributors/consumers. These log files are appended when licenses are served by the LS. If a consumer violates the system then there would be some mismatch between entries in the log files created at the consumer's machine and the permissions and constraints granted to the consumer. So, in order to detect violation, we need to compare the log files collected from the consumer's machine with the logs file at the LS related to the consumer. We also assume that the log files can be securely stored at consumers' machines, and any modification in the log files can be detected [6] [11]. If a distributor is involved in violation then there would be mismatch between the permissions and constraints granted to the distributor and granted by the distributor to other parties. So, in order to detect violation by the distributor, we need to compare entries involving the distributor in the log files created at the LS.

Logs from the LS and consumers are collected at the LCC for analysis purpose. Log analysis reports violation by an individual when at least one constraint with any permission is not satisfied. For example, if a consumer is allowed to play a content 10 times. But the logs collected from his machine show that content are played for more than 10 times then system should report violation. In the above example the violation is detected on the basis of the count based constraint. Where, the operator used for the count based constraints is 'summation'. Similarly, for detection of violation based on other types of constraints require other types of operators. For location based constraints violation the operator required would be 'subset'. i.e. for distributors, locations allowed in further licenses must be a subset of the locations in the parent licenses, and for consumers, location of content usage must be a subset of locations allowed in the usage licenses. For the time based constraints the operator required might be 'less than'(some times 'greater than' also). i.e. time of use of a permission must be less than the time permitted by the time based constraint associated with the permission.

So, to detect violation we need to select suitable data mining operations based on operators associated with different types of attributes. However, details about the data mining operations for different types of operators are out of scope of this paper.

V. Privacy Concerns

In DRM systems, data is collected for different purposes such as making future business strategies and violation detection. Data collected is very sensitive in DRM systems as it may disclose some facts about consumers that consumers do not want to disclose. Consider the following case: *a cancer patient may not want to disclose that he is a patient of cancer, but for his own benefit he purchases and views some documentaries on*

cancer. From the logs collected, it can be determined that the consumer is watching the documentaries on cancer and it may be concluded that he may be a patient of cancer. Thus, there is a likelihood that the privacy of the consumer is breached.

In the DRM system the owner/distributors are responsible for protecting privacy for their consumers. So usage data should be collected and processed by the owner/distributors for their own consumers. However, all distributors may not have enough resources to collect and process data. So, data is collected and processed by the LCC, which is under the control of the owner. Thus, it imposes a challenge on distributors to preserve privacy of their consumers. For this purpose, we use a temporary ID based scheme. In the scheme, distributors provide a temporary ID to their consumers. The temporary ID is used in usage logs in place of real ID. Logs at the LS should also contain only temporary ID and associated distributor as identifier. The LCC analyzes the logs for violation detection, if any violation is detected then the LCC may take help of the logs from the LS to identify distributor associated with temporary ID of consumer. Then the distributor concerned may be contacted to identify violators.

The scheme presented above is privacy preserving as the LCC cannot correlate the data collected with consumers'real ID. The data at the LS also does not contain real ID of the consumer. Only the concerned distributor/owner knows about the real ID and identity of consumers. So, identity of consumers can only be disclosed with the help of distributors.

VI. RELATED WORK

Liu et al. [1] reviewed general DRM systems involving the content provider, distributor, clearing house and consumer. Function of the distributor in all the DRM systems reviewed in [1] is similar to the function of the owner in our architecture. However, all of them support only two parties, whereas our proposed architecture supports multiparty (owner, distributors and consumers) and multiple levels of distributors. Hwang et al. in [4] presented digital rights models for various cases in DRM system including multilevel system. They assumed content distributors to be trusted by the content provider, and hence distributor can posses content keys. This limits number of distributors in their architecture as finding a large number of trusted distributors is very difficult in practice. In [2] DRM architecture for IPTV contents distribution based on peer-to-peer technology is proposed. But, in this architecture key management can be a major concern. For such a large system only one central key updating server may not be sufficient. With the help of multilevel structure the key management can be distributed between the owner, and distributors. Thus, it can reduce the burden on a single entity. Rosset et al. [3] described a DRM architecture with better security mechanism than basic DRM architectures described in [1]. But, contents in their architecture can only be distributed to authenticated users. So, super-distribution is not supported in their work and scalability is a major concern. A DRM architecture for content distribution using group ID concept is presented by Zhang et. al. in [7]. In the architecture a scheme for managing consumers and contents as hierarchical groups according to their features and granting rights to a group of consumers for group of contents is proposed. Although, their architecture is suitable for some scenarios like educational institutes but it is difficult to use it as a general DRM architecture because of difficulty in maintaining the user groups. The scenario in their work can be dealt efficiently using our multilevel architecture. Tradeoff between flexibility and security in DRM systems is discussed by Grimen et. al. [5]. Due to vulnerabilities most of the DRM systems are not fully protected against the attacks [5]. So, in addition to security mechanisms it is also necessary to detect any violation in the system. For this purpose, log files based violation detection method proposed in our architecture can be much effective.

VII. CONCLUSION

In this paper we have presented a multiparty, multilevel DRM architecture. The architecture takes care of all the parties involved in terms of content packaging and distribution, and license creation and distribution. The architecture also provides additional security by using log files based method for violation detection. The method is effective in case DRM system is violated. In the architecture, a privacy preserving mechanism is discussed to ensure that consumers' privacy does not breach because of usage data collection.

Acknowledgement: This work is supported by A*STAR, Singapore under the project 'Digital Rights Violation Detection for Digital Asset Management' (Project No: 0721010022).

REFERENCES

[1] Q. Liu, R. S. Naini, and N. P. Sheppard, *Digital Rights Management for Content Distribution*, Australian information security workshop, 2003.

[2] X. Liu, T. Huang, and L. Huo, *A DRM Architecture for Manageable P2P Based IPTV System*, IEEE Conference on Multimedia and Expo, pp. 899-902, July-2007.

[3] V. Rosset, C. V. Filippin, and C.M. Westphall, *A DRM Architecture to Distribute and Protect Digital Content Using Digital Licenses*, pp. 422-427, telecommunication, July-2005.

[4] S. O. Hwang, K. S. Yoon, K. P. Jun, K. H. Lee, *Modeling and implementation of digital rights*, Journal of Systems and Software, 73 (3), pp. 533-549, 2004.

[5] G. Grimen, C. Monch, and R. Midtstraum, *Building Secure Software-based DRM systems*, NIK 2006.

[6] J. E. Holt, *Logcrypt: Forward Security and Public verification for Secure Audit Logs*, Proceedings of the Australasian workshops on Grid computing and e-research, pp.203-211, January, 2006.

[7] J. Zhang, N Wu, J. Luo, and S.Yang, *A scalable Digital Rights Management Framework for Large Scale Content Distribution*, pp. 761-764, ISPACS, 2005.

[8] Y. Zheng, D. He, H. Wang, and X. tang, *Secure DRM Based Scheme for Future Mobile Networks Based on Trusted Mobile Platform*, Proceedings. International Conference on Wireless Communications, Networking and Mobile Computing, Volume 2, pp 1164 - 1167, 2005.

[9] S.Emmanuel, and M.S.Kankanhalli, *A Digital Rights Management Scheme for Broadcast Video*,ACM/Springer Multimedia Systems Journal, vol 8, no. 6, pp. 444 458, 2003.

[10] R. Hunt, *PKI and Digital Certification Infrastructure*, IEEE conference on networks, Pages: 234-239, Oct 2001.

[11] B. Schneier, and J. Kelsey, *Secure Audit Logs to Support Computer Forensics*, ACM Transaction on Information and System Security, pp. 159-176, Vol. 2, No. 2, May, 1999.

[12] *IFPI Music Report 2008*, available at: $http://www.ifpi.org/content/library/DMR2008.pdf$.