Key Management in Multi-Distributor based DRM System with Mobile Clients using IBE

Ratna Dutta
Claude Shannon Institute
NUI, Maynooth
Co. Kildare, IRELAND
ratna.dutta@nuim.ie

Sourav Mukhopadhyay
School of Electronic Engineering
Dublin City University
Dublin 9, IRELAND
msourav@eeng.dcu.ie

Tom Dowling
Claude Shannon Institute
NUI, Maynooth
Co. Kildare, IRELAND
tdowling@cs.nuim.ie

***Abstract*— Current Digital Rights Management (DRM) systems support only two-party systems, involving the package server and purchaser. However, for a scalable business model of transacting digital assets, a multi-party DRM system is often necessary which involves more than one distributors, who can promote and distribute the content in regions unknown to the package server. We propose a key management scheme for a DRM system that involves more than one distributors with the DRM client's flexibility of choosing a distributor according to his own preference. For instance, a mobile DRM client may contact to a distribution server who is nearest to him by location or who offers promotions/discounts on the price or offers more commissions. In our scheme, the package server does not trust the distribution servers or the license server. The encrypted digital content sent by a package server can only be decrypted by the DRM client who has a valid license and is protected from attacks by other parties/servers in the system. Moreover, we use Identity-Based Encryption (IBE) that incurs less computation cost and storage as certificate managements are not necessary and certificate verifications are no longer needed. These features make our DRM system suitable for more effective business models/applications with the flexibility in deciding a wide range of business strategies as compared to the existing works.**

***Index Terms*— DRM, key management, content protection, security.**

I. Introduction

DRM system is an important component of a digital asset management system, which manages the rights of the individuals involved in the creation and transaction of digital assets. The consumer purchases a digital license granting certain rights to him instead of buying the digital content. The content access is regulated with the help of license that contains permissions, constraints and content decryption keys. Permissions correspond to actions that can be performed on the contents, *e.g.* play, copy, edit, reuse and redistribute. Constraints are limitations associated with the permissions in the license, *e.g.* frequency of access and expiration date. Content decryption keys are used to decrypt encrypted contents and are available for a particular permission only if all the constraints associated with that permission are satisfied. For instance, suppose a license is issued with 'play' permission with constraints of 10 counts and validity of 30 days. The decryption key will be unavailable for play after 30 days even if less than 10 counts are used. These usage rules are often combined to enforce certain business models, such as rental or subscription, try-before-buy, pay-per-use and so forth.

Current DRM systems are mainly used for online music services, eBook publishing on PC-based platforms, games *etc.* With the widespread use of the Internet and improvements in streaming media and compression technology, DRM solutions found appealing applications in e-health to protect patients privacy. For example, it may be the case that doctors, pharmacists and nurses are required to have different rights to access and modify patients personal medical information over open network. Also in an online learning and information environment, a flexible and effective DRM solution facilitates trade and exchange of learning objects between universities/institutions on a free or fee basis by managing the creation, retrieval, trading and distribution of online learning objects and supporting collaborative development. A DRM system can also be used within a corporation to guarantee that only authorized people can access certain information and prevent employees from disclosing critical and proprietary information to the company's competitors.

Various DRM systems have been proposed for digital content and license distribution for a typical two-party scenario, where the owner and the consumer are the only parties involved in the system [20], [4], [23], [13], [7]. However, two-party DRM systems do not provide business scalability and unable to make proper business strategies. The DRM architectures in multi-party multi-level setup are addressed in [16], [21], [25], [33], [26]. Due to vulnerabilities, most of the DRM systems are not protected against the attacks. For instance, the solution presented in [16] assumes the distribution servers to be trusted by the owner and hence distributors can posses content keys. This is a shortcoming of the scheme as finding a large number of trusted distributors is very difficult. Authenticated key management and scalability are major concerns in multi-party multi-level DRM system. Trade-off between flexibility and security in DRM system is discussed in [12].

This article addresses the problem of designing a DRM architecture enabling proper business strategies for different regions and cultures, and designing an efficient and secure key management in this system. Our key management mechanism enjoys several interesting features as compared to the existing works. We summarise below our contributions in this paper

978-1-4244-4457-1/09/$25.00 ©2009 IEEE

and their advantages over the existing approaches:

1) We design a DRM system which is flexible to more innovative and scalable business model considering a network with multi-distributors instead of single-distributor. A local distributor can better explore potentially unknown market to the owner (package server) and make strategies according to the market. In addition, the distributors can also help in handling different price structure of media in different countries, and share with the owner any information on price or demand fluctuation cost. In our DRM system, the DRM client has the flexibility of choosing a distributor based on his own preference. The DRM client may be mobile and roam from one region to another. The DRM client may contact the distributor who is nearest to his location for a digital asset.

2) We provide a secure and efficient key management scheme in our proposed DRM system using IBE [30] instead of certificate-based Public Key Infrastructure (PKI), coupling it with Shamir's [29] secret sharing scheme. IBE has the property that a user's public key is an easily calculated function of his identity, such as his email address, while a user's private key can be calculated for him by a trusted authority, called Private Key Generator (PKG). The ID-based public key cryptosystem simplifies certificate management and certificate verification and is an alternative for certificate-based PKI, especially when efficient key management and moderate security are required. We obtain efficiency gains in computation time and storage over the existing certificate-based PKI approaches as no certificate management and certificate verification are needed by the entities in our DRM system.

3) In our key management mechanism, the package server does not trust distribution servers or license server. The symmetric decryption key used to encrypt a digital content is delivered from the package server to the DRM client in a secure manner and it is protected it from its generation to consumption. Unlike the current DRM system which has focused on content protection from purchasers, our scheme protects the key not only from the purchasers, but also from other principals such as the distribution servers and the license server. Consequently, the encrypted digital content sent by a package server can only be decrypted by the DRM client who has a valid license and no one else.

The rest of the paper is organized as follows: Section II presents the notations and terminologies used throughout the paper and briefly introduces the preliminaries on DRM systems, identity-based cryptography and digital signatures. In Section III, we propose our DRM model and key distribution scheme. The security analysis is provided in Section IV. Finally, we conclude in Section V.

II. Preliminaries

A. A Typical DRM System

Despite different DRM vendors have different DRM implementations, names and ways to specify the content usage rules, the basic DRM process is the same. The entities involved in a DRM system are a package server, distribution server, license server and DRM client [22], [18]. In this model, a purchaser is not a service provider, he simply pays a fee to the DRM client and watches a movie or listen to a song. Figure 1 displays the service/payment flow of a DRM system based on most existing commercial systems.

• **Package server:** The package server holds the digital rights of the content and wants to protect these rights. The package server is concerned about unauthorized usage (such as play, copy *etc.* without having permissions to perform) and illegal redistribution of contents. The package server's concern about unauthorized use of content is resolved by encrypting the content with the package server's own secret key. Digital contents have large volume and symmetric key is usually used to encrypt them as symmetric encryption provides high performance for consumption. For each digital content, a different symmetric key is used.

The package server's encryption key should not be disclosed to any party other than those who have the corresponding license (rights). The package server provides to the distribution server the encrypted content and content information (right metadata for the content promotion such as information to play the content, information about the compression algorithm *etc.*) The package server sends to the license server the encryption information such as the seed of the encryption key, the encryption length *etc.*

• **Distribution server:** The distribution server provides distribution channel such as online shop or a web retailer. The distribution server has a media server and sets up a website presenting the protected content and content information that he receives from the package server. A purchaser can select a content from the distribution server's website and download the encrypted content from its media server. Purchaser will be able to decrypt the content if it purchases the corresponding license from the DRM client which is issued to the DRM client by the license server.

• **License server:** The license server issues license to the DRM client when instructed by the distribution server. Digital licenses contain different permissions and usage rules such as frequency of access, expiration date, restriction of transfer to other devices, copy permission *etc.* Licence can be delivered to the requesting application prior to or at the same time as the transfer of digital content. Usually an e-commerce is integrated with a DRM system in handling financial payments and triggering the function of licence server. The license server handles the financial transactions for issuing the digital license to the DRM client, pays royalty fees to the package server and distribution fees to the distribution server accordingly. In this paper, we will not discuss the financial payment handling.

Moreover the license server is responsible to detect (or prevent) any unauthorized use due to system violation and take legal action against the DRM client. An effective method to detect unauthorized use due to system violation is by using log files that reflect actual activities of the DRM clients. Usage logs should be created at DRM client's machine and license server is responsible to collect and audit these logs and take necessary action if system violation is detected. We refer to [14], [28], [26] for more details on issues related to use of audit logs for detection/prevention of system violation. Digital watermarking [8], [9], [31], [24], tamper resistance [5], [27], [6], [1] are other means to detect system violation. We will

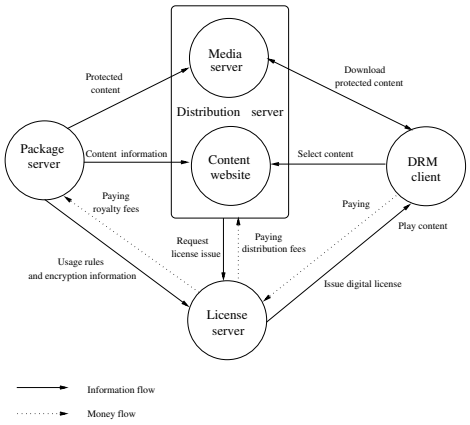

Fig. 1. Common components in DRM system

not describe the details here due to space consideration and concentrate mainly on key management among the different components of a DRM system.

• **DRM client:** The DRM client selects a content from the distribution server's web catalogue, uses the system to consume encrypted content by retrieving downloadable or streaming content through the distribution channel and then paying for the license. The DRM client analyzes the license (in which the decryption key is embedded) and decrypts the content. The DRM client provides service to the purchasers and purchasers are concerned about ease of getting content, ease of usage of content and their own privacy.

The following notations are used throughout the paper.

P	package server
D_i	i-th distribution server
L	license server
C	DRM client
ID_U	public identity of user U
S_{ID_U}	private key of user U
PKG	private key generator
Enc	ID-based asymmetric encryption algorithm
Dec	decryption algorithm corresponding to Enc
Sig	signature generation algorithm
Ver	signature verification algorithm
MK	master key of PKG
P_{pub}	public key of PKG
$A\|B$	concatenation of A and B

B. Certificate-Based Vs. Identity-Based Cryptography

The certificate-based protocols work by assuming that each entity has a static (long term) public/private key pair, and each entity knows the public key of each other entity. The static public keys are authenticated via certificates issued by a certifying authority (CA) by binding users' identities to static keys. When two entities wish to establish a session key, a pair of ephemeral (short term) public keys are exchanged between them. The ephemeral and static keys are then combined in a way so as to obtain the agreed session key. The authenticity of the static keys provided by signature of CA assures that only the entities who posses the static keys are able to compute the session key. Thus the problem of authenticating the session key is replaced by the problem of authenticating the static public keys which is solved by using CA, a traditional approach based on a Public Key Infrastructure (PKI).

However, in a certificate-based system, the participants must first verify the certificate of the user before using the public key of the user. Consequently, the system requires a large amount of computing time and storage.

In identity-based public key encryption, the public key distribution problem is eliminated by making each user's public key derivable from some known aspect of his identity, such as his email address. When Alice wants to send a message to Bob, she simply encrypts her message using Bob's public key which she derives from Bob's identifying information. Bob, after receiving the encrypted message, obtains his private key from a third party called a Private Key Generator (PKG), after authenticating himself to PKG and can then decrypt the message. The private key that PKG generates on Bob's query is a function of its master key and Bob's identity.

Shamir [30] introduced this concept of identity-based cryptosystem to simplify key management procedures in certificate-based public key infrastructure. The first ID-Based Encryption (IBE) was proposed by Boneh and Franklin [3] in 2001 that uses bilinear pairing. Shortly after this, many ID-based cryptographic protocols were developed (see [10] for a survey) based on pairings and is currently a very active area of research.The ID-based public key cryptosystem can be an alternative for certificate-based PKI, especially when efficient key management and moderate security are required.

The advantage of ID-based encryption are compelling. It makes maintaining authenticated public key directories unnecessary. Instead, a directory for authenticated public parameters of PKGs is required which is less burdensome than maintaining a public key directory since there are substantially fewer PKGs than total users. In particular, if everyone uses a single PKG, then everyone in the system can communicate securely and users need not to perform on-line lookup of public keys or public parameters.

In an ID-based encryption scheme there are four algorithms.

1) Setup : Creates system parameters and *master key*.
2) Extract : Uses master key to generate the private key corresponding to an arbitrary public key string ID.
3) Encrypt : Encrypts messages using the public key ID.
4) Decrypt : Decrypts the message using the corresponding private key of ID.

C. Digital Signature

Digital signatures are one of the most important cryptographic primitives. In traditional public key signature algorithms, the binding between the public key and the identity of the signer is obtained via a digital certificate. Shamir [30] first noticed that it would be more efficient if there was no need for such bindings, in that case given the user's identity, the public key could be easily derived using some public deterministic algorithm. This makes efficient ID-based signature schemes desirable. In ID-based signature schemes, verification function

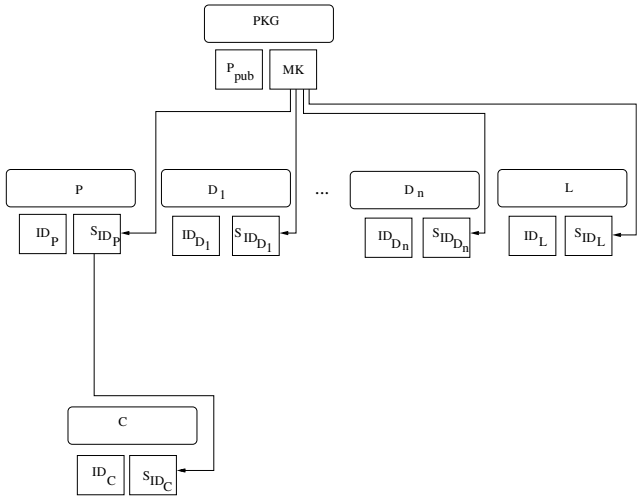

Fig. 2. Asymmetric key distribution: PKG issues private keys for package server P and distribution servers $D_1, \ldots, D_n$, whereas package server P issues private key for DRM client C

is easily obtained from the identity, possibly the same key and the same underlying computation primitives can be used. Shamir gave a practical ID-based signature scheme.

A standard digital signature scheme $(\mathsf{KeyGen}, \mathsf{Sig}, \mathsf{Ver})$ consists of three algorithms.

1) KeyGen : the key generation algorithm that generates randomly public system parameters params and public/secret key pair PK, SK of a signer.
2) Sig : signature generation algorithm that generates a signature on a given message m using the secret key SK of a signer.
3) Ver : signature verification algorithm that checks the validity of a signature on a given message using the public key of a signer and returns true or false as the case may be.

For ID-based signature, a signer's public key PK is its public identity and secret key SK is the key that the signer obtains by extract query on its identity to PKG.

III. Proposed Key Distribution

A. Asymmetric Key Distribution

The commonly used cryptographic primitives in DRM systems are symmetric and asymmetric encryption, digital signatures, one way hash functions, digital certificates *etc.* To mitigate the bandwidth overhead, among several asymmetric (public) key cryptography one may adopt Elliptic Curve Cryptography (ECC) [2] due to its acceptable overhead. The signature scheme ECC-192 provides higher security level than RSA-1024 while the length of its signature is 48 bytes compared to 128 bytes of RSA-1024. In our asymmetric key distribution, we use the setup of Identity-Based Encryption (IBE) instead of certificate-based setup to simplify certificate management and certificate verification. A trusted PKG generates the private key of a server upon receiving its public identity (which may be some known aspect of its identity, such as its e-mail address or biometric). We use the private/public key pair thus generated for each entity in the system as the respective signing/verification key pair of the corresponding entity.

In our DRM model, the package server P appoints n distribution servers $D_1, \ldots, D_n$ in different regions to facilitate the distribution process. The DRM client C is mobile and moves from one region to another. C can download encrypted contents from its preferred distributor, say D_i, which might be location wise nearest to C. The owner of the package server P has raw content and wants to protect it. None of the principals except P should know how to decrypt the content. Our proposed key management scheme deals with the key management among several components of a DRM system. The main ideas are the followings:

- Symmetric encryption is used to encrypt digital content by the package server P.
- Partial information of symmetric decryption keys are delivered using asymmetric encryption and stored in different servers in such a way that neither the distribution servers $D_1, \ldots, D_n$ nor the license server L can generate the decryption key.
- The components of a DRM system which have a content decryption key are the package server P and the DRM client C with a valid license.

It is very difficult to authenticate a purchaser. Purchases are concerned about their privacy and anonymity. They simply needs to pay a fee to watch a movie. Instead, the DRM client C is a service provider to the purchaser and should be authenticated by the owner of the package server P. Figure 2 shows the key distribution of asymmetric keys which are used to deliver symmetric decryption keys and mutually authenticate the components of a DRM system.

The principals of the package server P, the distribution servers $D_1, \ldots, D_n$ and the license server L submit their respective public identities to PKG and receive the corresponding secret keys through a secure communication channel, after PKG verifies the identities of the principals. PKG uses its master key and received valid identity of a principal to generate the principal's corresponding private key. The package server P plays the role of PKG for the DRM client C and issues its private key in a secure manner after verifying the public identity of C. *i.e.* P, $D_1, \ldots, D_n$ and L make Extract query on their respective identities to PKG, whereas C makes Extract query on its identity to the package server P. P uses its own private key issued by PKG to compute the private key of the DRM client C corresponding to C's public identity.

B. Key Delivery when Packaging

While packaging a digital content M, the package server P uses a symmetric key K to encrypt M and delivers partial information of K to the license server and n distribution servers $D_1, \ldots, D_n$ in the following manner. The service flow is shown in Figure 3.

1)(a) P first chooses a polynomial $f(x) \in F_q[x]$ of degree t with $K = f(0)$, where F_q is a finite field of a large prime order q.

(b) P computes for $1 \leq i \leq n$, $Y_{D_i} = \mathsf{Enc}_{\mathsf{ID}_{D_i}}(f(i))$ using D_i's public identity ID_{D_i}, generates signature $\sigma_{Y_{D_i}} =$

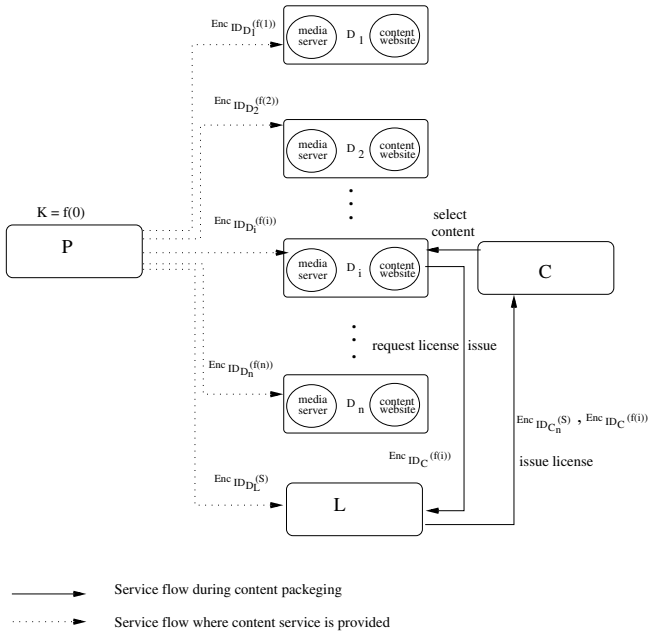

Fig. 3. Key delivery mechanism (signature and verification are not shown), where $S = \{(x_i, f(x_i)) : 1 \leq i \leq t\}$ and $x_1, \ldots, x_n \in Z_q^* \setminus \{1, \ldots, n\}$ and D_i is the preferred distribution server (location wise nearest) to the DRM client C.

$\mathsf{Sig}_{S_{\mathsf{ID}_P}}(Y_{D_i})$ using P's own private key S_{ID_P} and sends $Y_{D_i}|\sigma_{Y_{D_i}}$ to D_i.

(c) P chooses randomly t distinct elements $x_1, \ldots, x_t \in Z_q^* \backslash \{1, \ldots, n\}$. P computes $Y_L = \mathsf{Enc}_{\mathsf{ID}_L}(S)$ using L's public identity ID_L where $S = \{(x_i, f(x_i)) : 1 \leq i \leq t\}$, signature $\sigma_{Y_L} = \mathsf{Sig}_{S_{\mathsf{ID}_P}}(Y_L)$ using P's own private key S_{ID_P}, and sends $Y_L|\sigma_{Y_L}$ to L.

2)(a) For $1 \leq i \leq n$, D_i on receiving $Y_{D_i}|\sigma_{Y_{D_i}}$, verifies the signature $\sigma_{Y_{D_i}}$ on Y_{D_i} using P's public identity ID_P. If verification succeeds, *i.e.* $\mathsf{Ver}_{\mathsf{ID}_P}(Y_{D_i}, \sigma_{Y_{D_i}}) = \mathsf{true}$, then D_i decrypts Y_{D_i} using its private key $S_{\mathsf{ID}_{D_i}}$, recovers $f(i) = \mathsf{Dec}_{S_{\mathsf{ID}_{D_i}}}(Y_{D_i})$ and stores $f(i)$ to its secure database.

(b) L upon receiving $Y_L|\sigma_{Y_L}$, verifies the signature σ_{Y_L} on Y_L using P's public identity ID_P. If verification succeeds, *i.e.* $\mathsf{Ver}_{\mathsf{ID}_P}(Y_L, \sigma_{Y_L}) = \mathsf{true}$, then L decrypts Y_L using its private key S_{ID_L}, recovers $S = \mathsf{Dec}_{S_{\mathsf{ID}_L}}(Y_L)$, where S is the set of points $S = \{(x_i, f(x_i)) : 1 \leq i \leq t\}$ and stores them to its secure database.

C. Key Delivery when Content Service is Provided

When a DRM client C requests the content service for encrypted content M from a distribution server, say D_i, which is within nearest reach to C, the following steps are executed. Figure 3 displays the service flow.

1) D_i computes $Y_C = \mathsf{Enc}_{\mathsf{ID}_C}(f(i))$ using C's public identity ID_C, signature $\sigma_{Y_C} = \mathsf{Sig}_{S_{\mathsf{ID}_{D_i}}}(Y_C)$ using D_i's own private key $S_{\mathsf{ID}_{D_i}}$, and sends $Y_C|\sigma_{Y_C}$ to L.

2) L on receiving $Y_C|\sigma_{Y_C}$, verifies the signature σ_{Y_C} on Y_C using D_i's public identity ID_{D_i}. If verification succeeds, *i.e.* $\mathsf{Ver}_{\mathsf{ID}_{D_i}}(Y_C, \sigma_{Y_C}) = \mathsf{true}$, L computes $Y_L = \mathsf{Enc}_{\mathsf{ID}_C}(S)$ using C's public identity ID_C, signature $\sigma_{Y_C|Y_L} = \mathsf{Sig}_{S_{\mathsf{ID}_L}}(Y_C|Y_L)$ using L's own private key S_{ID_L}, and issues the license that contains $Y_C|Y_L|\sigma_{Y_C|Y_L}$ together with rights, content URL, and so forth.

3) The DRM client C analyzes the licence issued by L, verifies $\sigma_{Y_C|Y_L}$ on $Y_C|Y_L$ using L's public key ID_L. If verification succeeds, C decrypts Y_C and Y_L using its own private key S_{ID_C}, and recovers $f(i) = \mathsf{Dec}_{S_{\mathsf{ID}_C}}(Y_C)$ and $S = \mathsf{Dec}_{S_{\mathsf{ID}_C}}(Y_L)$ where $S = \{(x_i, f(x_i)) : 1 \leq i \leq t\}$. C then interpolates with S and $(i, f(i))$ to recover $K = f(0)$ by Lagrange interpolation formula as follows:

$$f(0) = \sum_{l=0}^{t} \Lambda_l f(x_l),$$

where

$$\Lambda_l = \prod_{\substack{k=0 \\ k \neq l}}^{t} \frac{-x_k}{x_l - x_k}$$

with $x_0 = i$. Finally, C decrypts the content using the recovered symmetric key K and can view (play) M.

IV. Security

The process of authentication or verification of the identities of the parties is necessary in a DRM system to ensure that the packaged digital content is from the genuine authorized content distributor. In our design, digital certificates are not used to authenticate or verify the identity of the parties involved in the system unlike certificate-based public key infrastructure, thus saving large amount of computing time and storage. Instead, we use IBE that simplifies our key management mechanism. An attack on the $(n+1)$ partial information of the symmetric decryption key K (which is used in encryption for content protection by the package server) during delivery from the package server P to the distribution servers $D_1, \ldots, D_n$ and the license server L is prevented, because each piece of the $(n+1)$ partial information of K is encrypted under a public key and delivered to a server who owns the matching private key. Note that to recover the decryption key, one needs to know $(t+1)$ points on the polynomial $f(x)$. The $(n+1)$ partial information of K are separated and stored at different servers in such a way that, neither any of the distribution servers $D_1, \ldots, D_n$ nor the license server L has $t+1$ points on the polynomial $f(x)$ to generate the decryption key $K = f(0)$ by itself. Hence the decryption key K is protected from an attack on the distribution servers or the license server, since the $(n+1)$ partial information of K is stored at different servers so that each server knows insufficient points on the polynomial $f(x)$ to interpolate it and get the key $K = f(0)$.

Moreover, since a distribution server encrypts its partial information of K with the DRM client's public key and sends it to the license server, the license server cannot decrypt it and consequently, cannot generate the decryption key K. License server also encrypts its partial information of K using the DRM client's public key. Thus the partial information of K can only be decrypted by the DRM client who has the matching private key and no one else. The DRM client gets $(t+1)$ points on the polynomial $f(x)$ after decryption and combine them to recover the key $K = f(0)$ by Lagrange interpolation.

Our key management scheme enables the symmetric decryption key K to be protected from the principals who manages the distribution servers and the license server. The digital content can thus be protected from attacks during the content distribution since the encrypted digital content is sent by the package server and only the DRM client can decrypt the digital content. Besides, we use IBE and digital signature instead of digital certificates. This simplifies the process of authentication or verification of the identities in the system.

We use digital signatures for non-repudiable rights issuing. The license server digitally signs licenses of the digital content. Consequently, the play application on the DRM client's device can verify the correctness of the usage rights and keep the signature as a proof of rights purchase. One can combine one-way hash functions such as HMAC-SHA1 [19] in the DRM system with digital signature for integrity checking. The license server uses its private key to sign the hash value of the encrypted content rights. Integrity verification of the license is through verifying the signature using the public key of the license server and then comparing the hash value with a re-computed hash value. Similar arguments hold for the other servers (the package server and the distribution servers in the system).

V. Conclusion

In this paper, we present a flexible and effective DRM architecture with multi-distributors that facilitates client mobility and an efficient key management mechanism in this DRM system coupling IBE with Shamir's secret sharing. Our proposed DRM architecture provides scalability of business model and allows to make proper business strategies for different regions and cultures. The encrypted digital content sent by a package server can only be decrypted by the DRM client and is protected from attacks by other parties/servers in our DRM system. Our key management protects the key used to encrypt a digital content during its delivery from the package server to the DRM client, not only from purchasers but also from the distribution servers and the license server. IBE enables us to obtain efficiency gains in computation time and storage over the existing certificate-based PKI approaches as no certificate management and certificate verification is needed by the entities in our DRM system.

References

[1] Alchemedia. *Technology Benefits*. 2002. http://www. alchemedia.com/benefits/technology.html.

[2] *ANSI X9.62, Public Key Cryptography for the Financial Services Industry*. The Elliptic Curve Digital Signature Algorithm, 1999.

[3] D. Boneh and M. Franklin. *Identity-Based Encryption from Weil Pairing*. In proceedings of Crypto 2001, LNCS 2139, pages 213-229, Springer-Verlag, 2001.

[4] L. J. Camp. *First Principles of Copyright for DRM Design*. IEEE Internet Computing, vol.7, pp. 59-65, May-June 2003.

[5] H. Chang, and M. G. Atallah. *Protecting Software Code By Guards*. ACM Workshop on Security and Privacy in Digital Rights Management 2001. Pennsylvania, USA.

[6] Cloakware Corporation. *Protecting digital content using cloakware code transformation technology*. Cloakware Whitepapers, 2002. http://www.cloakware.com/ resources/.

[7] J. E. Cohen. *DRM and Privacy*. Communications of the ACM, vol. 46, issue 4, Apr. 2003.

[8] I. J. Cox, and M. L. Miller. *A review of watermarking and the importance of perceptual modeling*. Proc. of Electronic Imaging 97, February 1997. NEC Research Institute, 1997.

[9] J. Dittmann, P. Wohlmacher, and R. Ackermann. *Conditional and User Specific Access to Services and Resources using Annotation Watermarks.* Communications and Multimedia Security Issues of The New Century. pp.137-142. Ralf Steinmetz, Jana Dittman and Martin Steinebach (eds). Kluwer Academic Publishers, 2001.

[10] R. Dutta, R. Barua and P. Sarkar. *Pairing Based Cryptographic Protocols : A Survey*. Manuscript 2004. Available at http://eprint.iacr.org/2004/064.

[11] 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.

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

[13] F. Hartung and F. Ramme. *Digital Rights Management and Watermarking of Multimedia Content for M-Commerce Applications*. IEEE Comm., vol. 38, pp.78-84, Nov. 2000.

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

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

[16] 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.

[17] *IFPI Music Report 2008*, available at: http://www.if pi.org/content/library/DM R2008.pdf .

[18] Y. Jeong, K. Yoon, J. Ryou. *A Trusted Key Management Scheme for Digital Rights Management*. ETRI Journal, Vol. 27, No. 1, pp. 114-117, Feb. 2005

[19] Krawczyk, H., Bellare, M. and R. Canetti. *HMAC: Keyed- Hashing for Message Authentication*. RFC 2104, Feb. 1997.

[20] J. Lee, S. Hwang, S. Jeong, K. Yoon, C. Park, and J. Ryou. *A DRM Framework for Distribution Digital Contents through the Internet*. ETRI Journal, vol. 25, pp.423-436, Dec. 2003.

[21] 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.

[22] Q. Liu, R. Safavi-Naini, and N. P. Sheppard. *Digital Rights Management for Content Distribution*. Proceedings of Australasian Information Security Workshop Conference on ACSW Frontiers 2003, vol. 21, Jan 2003.

[23] D. K. Mulligan, J. Han, and A. J. Burstein. *How DRM- Based Content Delivery Systems Disrupt Expectations of Personal Use*. Proc. 2003 ACM Works. Digital Rights Management, pp.77-88, Oct. 2003.

[24] F. A. P. Petitcolas, R. J. Anderson, and M. G. Kuhn. *Attacks on Copyright Marking Systems. David Aucsmith*. Ed., Second workshop on information hiding, in vol. 1525 of Lecture Notes in Computer Science, Portland, Oregon, USA, pp. 218-238, 1998.

[25] 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.

[26] A. Sachan, S. Emmanuel, A. Das, M. S. Kankanhalli. *Privacy Preserving Multiparty Multilevel DRM Architecture*. IEEE Consumer Communications and Networking Conference (CCNC), Jan. 2009.

[27] T. Sander and C. F. Tschudin. *Protecting Mobile Agents Against Malicious Hosts*. In G. Vigna (ed.), Mobile Agent Security, LNCS, 1998.

[28] 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.

[29] A. Shamir. *How to Share a Secret*. Communications of the ACM, Vol. 22, No. 11, pp. 612-613, 1979.

[30] A. Shamir. *Identity-based Cryptosystems and Signature Schemes*. In proceedings of Crypto 1984, LNCS 196, pp. 47-53, Springer, 1984.

[31] M. Stamp. (2002). *Digital Rights Management: The Technology Behind the Hype*. Cupertino, CA, 2002. http://home.earthlink.net/ mstamp1/papers/DRMpaper. pdf.

[32] M. Valimaki and O. Pitkanen. *Digital Rights Management on Open and Semi-Open Networks*. Proc, WIAPP 2001, pp.154-155, July 2001.

[33] 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.