Secure Multimedia Content Delivery with Multiparty Multilevel DRM Architecture

Tony Thomas, Sabu Emmanuel, Amitabha Das
School of Computer Engineering
Nanyang Technological University, Singapore
{ttony,asemmanuel,asadas} @ntu.edu.sg

Mohan S. Kankanhalli
School of Computing
National University of Singapore, Singapore
mohan@comp.nus.edu.sg

ABSTRACT

For scalability of business, multiparty multilevel digital rights management (DRM) architecture, where a multimedia content is delivered by an owner to a consumer through several levels of distributors has been suggested as an alternative to the traditional two party (buyer-seller) DRM architecture. A combination of cryptographic and watermarking techniques are usually used for secure content delivery and protecting the rights of seller and buyer in the two party DRM architecture. In a multiparty multilevel DRM architecture the cryptographic and watermarking mechanism need to ensure the secure delivery of the content as well as the security concerns of the owner, multiple levels of the distributors and the consumer. In this paper, we propose a mechanism which takes care of the above security issues, for delivering multimedia content through multiparty multilevel DRM architecture.

Categories and Subject Descriptors

J.7 [**Computers in Other Systems**]: Consumer Products

General Terms

Algorithms, Design, Security

Keywords

Digital rights management, Watermarking, Chinese remainder theorem

1. INTRODUCTION

With the high proliferation of Internet, it has become very easy to obtain, replicate and distribute digital content without any loss of quality. This has resulted in the illegal replications and distributions of digital content at ease causing widespread violations of intellectual property rights. Hence DRM (Digital Rights Management) technologies have been developed using encryption and digital watermarking techniques to prevent consumers from unauthorized copying of digital content, to control the use of digital content, and to enable the development of digital distribution platforms on which innovative business models can be implemented. Encryption prevents unauthorized access to digital content, but once a content is decrypted, it doesn't prevent an authorized user from illegally replicating the digital content. Digital watermarking is used to complement the encryption techniques, to establish and prove ownership rights and to trace copyright violators by embedding the seller's and buyer's information into the media.

The traditional two party digital rights management architecture involving a seller and buyer is not adequate to satisfactorily address the requirements of the present day business models for content delivery. Hence, multiparty multilevel digital rights management architecture (MPML-DRM-A) has been used as an alternative to the two party (buyer-seller) DRM architecture by many authors [3, 12]. The term multiparty refers to the multiple parties such as the owner, distributors, sub-distributors and consumers and multilevel refers to the multiple levels of distributors/sub-distributors involved in the distribution chain of a content. In multiparty multilevel DRM architecture, the encryption and watermarking mechanism need to ensure that the security concerns of all the parties are taken care.

In a multiparty multilevel DRM architecture, the content can be securely passed from the owner to the consumer through distributors by encrypting the content. However, protecting the security concerns of all the involved parties through watermarking is not easy. If each party embeds its watermark signal separately into the digital content, the quality of digital content will deteriorate with each watermarking. Therefore, how to protect the rights of the owner, distributors and consumer through watermarking is a very important issue in this architecture. In this paper, we propose a joint digital watermarking mechanism for this architecture for protecting the rights of all parties. All the parties involved jointly generate a joint watermark information using Chinese remainder theorem (CRT) and finally the DRM agent at the consumer machine generates a watermark signal out of it and embeds into the content. In the event of finding an illegal copy the traitors can be traced.

The remaining of the paper is organized as follows. The preliminaries are given in Section 2. In Sections 3, our joint watermarking mechanism is given. The paper concludes with remarks and future directions for research in Section 4.

2. PRELIMINARIES

In this section, we discuss the preliminaries required.

NOSSDAV'09, June 3–5, 2009, Williamsburg, Virginia, USA.

2.1 Notations

We start with the notations we will be following.

- Entities involved are: owner O, k distributors $D_1, \ldots, D_k$, consumer C, license server L and judge J.
- X denotes the content and l_X the copy number of X.
- $H(.)$ denotes any standard hash function such as SHA1 or MD5 and $||$ denotes the concatenation operator.
- $E_{pub}(.|K)$, $D_{pub}(.|K)$, $Sig(.|K)$ and $Ver(.|K)$ denote encryption, decryption, digital signature generation and digital signature verification algorithms (key K) respectively, corresponding to a public-key cryptosystem.
- For $i = 0, \ldots, k+1$,
  - (e_i, d_i) denotes the public-private-key pairs, $Cert_i$ denotes the public-key certificate and r_i denotes the watermark information

  of O, $D_1 \ldots, D_k$ and C respectively.
- For $i = 0, \ldots, k$,
  - n_i denotes a prime number (all are distinct), CS_i denotes the content server, UL_i and RdL_i denote the usage and redistribution licenses

  of O, $D_1 \ldots, D_k$ respectively.
- $E_{sym}(.|K)$, $D_{sym}(.|K)$ denote the encryption and decryption algorithms corresponding to a standard symmetric key cryptosystem like AES or 3DES.
- $E_{sym}(x_1, \ldots, x_p|K) = (E_{sym}(x_1|K), .., E_{sym}(x_p|K))$. $D_{sym}(y_1, \ldots, y_p|K) = (D_{sym}(y_1|K), .., D_{sym}(y_p|K))$.
- I denotes a joint watermark information and W denotes the corresponding joint watermark signal.
- $W_{gen}(.|K)$, $W_{emb}(.|K)$ and $W_{det}(.|K)$ denote any standard watermark signal generation algorithm, robust watermark embedding algorithm and the watermark detection algorithm (with key K) respectively.
- K_X and K'_X denote the keys used for embedding and detecting the watermark signal in X respectively.

2.2 Multiparty Multilevel DRM Architecture

To efficiently deliver multimedia content by proxy caching of media content, many proxy-based media distribution systems have been proposed [1]. Multiparty multilevel DRM architecture (MPML-DRM-A) is a more general framework involving many parties like owner, multiple levels of distributors (including none), consumers and a license server. Some DRM systems involving these parties is given in [7]. The function of the distributor in these systems is just similar to that of the owner and is not really a general setup. The DRM architecture for IPTV content distribution given in [6] has issues with key management. The DRM architecture described in [11] does not support super-distribution and is not scalable. The DRM architecture given in [13], uses a group ID concept and is difficult to use as a general DRM architecture. The multiparty DRM architecture proposed in [12] with its multilevel structure is more general and takes care of the many limitations of the above architectures. Hence, we adopt this DRM architecture for our discussion. A schematic diagram of this architecture is given in Fig 1. The content moves from the owner to the consumer through multiple levels of distributors. The owner and distributors maintain their own content servers CS. The license server issues redistribution licenses to distributors and usage licence to the consumers. A redistribution license allows the receiver to redistribute the content and usage licence allows the receiver to use the content. A license grants the receiver specific permissions, constraints and content decryption keys.

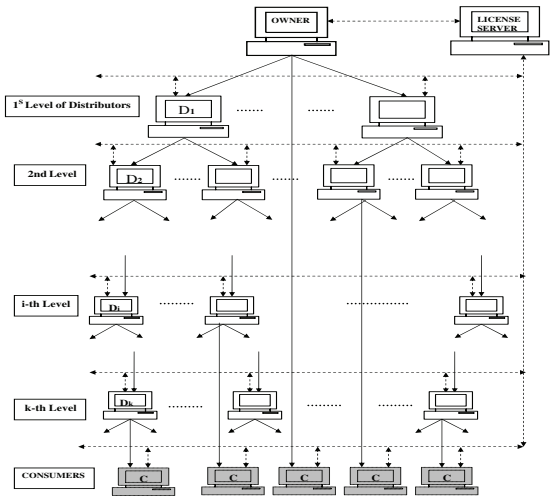

Figure 1: This figure shows the multiparty multilevel DRM architecture. The dark arrows show the flow of the content and dotted arrows show the communication between an entity with the license server. As the figure shows a consumer can get the content from owner or any distributor at any level.

The following section discusses the security concerns other than secure content delivery with MPML-DRM-A.

2.3 Security Concerns in MPML-DRM-A

If a distributor or owner finds an illegal copy of the content, he should be able to *trace the traitor* responsible for it. Innocent parties who were not involved in a content delivery should be protected against *false framing* by others. The *consumers and distributors* should be protected against false allegations by a super-distributor or owner regarding illegal distribution of a content. Once a content has been delivered to a consumer, there should be a mechanism for the owner and the distributors involved to *prove their role*. Malicious parties should not be able to *collude* with each other and mount attacks against other parties such as owner or distributors or consumers. A consumer should be able to get a content anonymously and the *privacy of the consumer* should be preserved until a need arises to trace the traitor in the event of finding an illegal copy of the content.

These security concerns can be taken care by embedding a watermark signal created jointly by all the parties involved. We do this joint watermark creation using CRT.

2.4 Secure Delivery of Multimedia Content

A multimedia content can be securely delivered to a consumer in MPML-DRM-A through simple cryptographic mechanisms as described in [12]. The content encryption scheme is based on a global encryption key (GEK) and a set of local encryption keys (LEK). GEK prevents unauthorized use of contents, and LEK's prevent illegal download of the content from the content servers. The owner first encrypts the content X with global encryption key GEK and then encrypts the initially encrypted content with his local encryption key LEK_0 using a symmetric key encryption algorithm and uploads the result on his content server. A distributor D_i downloads the content from the content server of the owner or a higher level distributor. It then obtains the re-distribution license from the licence server. The distributor is allowed to decrypt the content using LEK_{i-1} of the owner/distributor. Thus, the distributor gets only content encrypted with GEK. It then encrypts the above encrypted content with its local encryption key LEK_i. The distributor then uploads the result on its content server. A consumer downloads the content from the content server of any distributor or owner. It then obtains the usage license for the content from the license server. A trusted DRM agent at the consumer's device can decrypt the content using LEK of the distributor concerned as well as GEK of the owner.

2.5 Key Generation and Distribution

Each entity O, D_i and L obtain their public-key cryptographic credentials (public and private key pairs (e_i, d_i) and digital certificate $Cert_i$) through a public-key infrastructure (PKI). The owner generates the keys GEK and LEK_0 randomly, digitally signs them and uploads the digital signatures of the keys along with the encrypted content in the content server CS_0. Similarly the i-th level distributor D_i generates its local encryption key LEK_i randomly, digitally signs it and uploads the digital signature of the key along with the encrypted content in the content server CS_i.

2.6 Privacy Preserving Content Download

In this section, we discuss a mechanism described in [1] adapted to our MPML-DRM-A domain for protecting the privacy of a consumer. A consumer C can access a content without letting the content server know which multimedia content the consumer is accessing. The details are as follows.

An encryption function E is said to be a *commutative* if given two encryption keys e_1 and e_2 and a message m,

$$E(E(m|e_1)|e_2) = E(E(m|e_2)|e_1). \quad (1)$$

That is the order of keys used in encryption does not matter. Pohlig-Hellman based on elliptic curve cryptography and Shamir-Omura based on RSA are examples of commutative encryptions [1]. If $D(.)$ is the decryption function corresponding to $E(.)$, then as a consequence of Equation (1), if $c_2 = E(m|e_2)$ and $c = E(E(m|e_1)|e_2)$, we have $c_2 = D(c|e_1)$. We now describe the privacy preserving content download.

The license server L and the consumer C generate key pairs (e_L, d_L) and (e_C, d_C) respectively based on Shamir-Omura cryptosystem. Each content X is associated with a unique identity ID_X. The consumers can get the identities ID_X (for various X) from the content server of distributors or publicly. However, the association of an encrypted content with a content identity will not be known to content servers of the distributors. The (encrypted) contents kept in a content server are identifiable through only their encrypted content identities. The initial setup is as follows.

- The owner passes the content identity ID_X of a multimedia content X to the license server L.
- L encrypts ID_X with its key as $ID_X^L = E(ID_X|e_L)$.
- Owner and distributors upload encrypted contents and encrypted content identities in their content servers.

Now the protocol for secure content download is as follows:

1. Consumer C encrypts ID_X with e_C to get $ID_X^C = E(ID_X|e_C)$ and sends to the licence server L.
2. L encrypts ID_X^C with e_L to get $ID_X^{CL} = E(ID_X^C|e_L) = E(E(ID_X|e_C)|e_L)$ and sends to C.
3. C decrypts ID_X^{CL} to get $ID_X^L = D(ID_X^{CL}|d_C)$.
4. Since the multimedia content X is identifiable in the content server via ID_X^L, C can download the content from the server without letting the owner, distributor and the license server know which content it accessed.

2.7 CRT and Related Works

The Chinese Remainder Theorem (CRT) is as follows. Let $n_1, \ldots, n_k$ are pairwise coprime positive integers and $r_1, \ldots, r_k$ are any set of integers. Then the k congruences

$$x \equiv r_i \mod n_i, \text{ for } 1 \leq i \leq k,$$

has a unique solution x such that $0 \leq x < N = n_1 \ldots n_k$ [9].

There exists several joint watermarking mechanisms [8, 4] for the two party DRM architecture. However, there has not been any work on joint watermarking for multiparty multilevel DRM architecture yet. Our CRT based joint watermarking scheme seems to be the first in this direction.

CRT has been used in many DRM applications. Some such applications are a key distribution scheme for conditional access system in digital TV broadcast [2], a CRT based parameter distribution in the scrambling process for conditional access to Pay-TV systems [5] and a binary fingerprinting code using CRT [10].

3. PROPOSED WATERMARKING SCHEME

In this section, we describe our joint watermarking protocol based on CRT for MPML-DRM-A. To make the presentation simpler, we have not explicitly included the secure content delivery mechanism described in Section 2.4 and privacy preserving content download protocol described in Section 2.6 although it is assumed in the complete framework.

The proposed watermarking protocol involves the following entities: owner O (D_0), k levels of distributors $D_1, \ldots, D_k$ ($k \geq 0$), a consumer C and a license server L.

We generate the joint watermark information I as the (CRT) solution of a set of congruences corresponding to each party in the distribution chain. The watermark signal W is generated from this joint watermark information using a watermark generation algorithm and then embedded into the content using a robust embedding algorithm. The watermark signal is detected using the corresponding watermark detection algorithm. Due to space constraints, we do not elaborate on the watermark signal generation, embedding and extraction algorithms in this paper.

3.1 Individual Watermark Information

Each party i (owner/distributor/consumer) involved generates its individual watermark information r_i using its private key d_i as its digital signature $r_i = Sig(H(H(X)||l_X)|d_i)$.

The following section describes how the individual watermark informations r_i are combined to generate a joint watermark information I using Chinese remainder theorem.

3.2 Joint Watermark Information Generation

Let $r_0, r_1, \ldots, r_k$ and r_{k+1} be the individual watermark information of the parties $O, D_1, \ldots, D_k$ and C respectively, computed as described in the Section 3.1. Let $n_0, n_1, \ldots, n_{k+1}$ be prime numbers of same size assigned to these parties respectively. Then their joint watermark information I is the solution of the following set of $k+2$ congruences:

$$I \equiv r_i \mod n_i, \quad \text{where} \quad i = 0, 1, \ldots, k+1. \tag{2}$$

The existence and uniqueness of I is guaranteed by CRT.

3.3 Joint Watermark Embedding Protocol

Recall all the notations given in Section 2.1. Let a multimedia content reaches a consumer C from the owner O through k distributors $D_1, \ldots, D_k$. We now describe the watermark embedding protocol below.

We begin with the interactive protocol and the computations performed by the owner O with the license server L. L first establishes a session key K_0 with O. All the further secure communication between them are performed with symmetric key cryptography using key K_0. O computes its watermark information r_0, its re-distribution license RdL_0 and usage license UL_0 and sends to L after encryption with K_0. L decrypts the encrypted licenses using K_0 and verifies the licenses and r_0. If all the verifications pass through, L clears O to upload the encrypted content on its content server CS_0. O generates a unique watermark signal W_{own} which is a function of the copy identifier l_X and embeds into the content X using any robust watermarking algorithm to get X' and then encrypts and uploads on CS_0. Formally the steps are as follows.

1. O sends $Cert_0$ to the Licence server L.
2. L verifies $Cert_0$, extracts e_0, generates a random session key K_0 and sends $K' = E_{pub}(K_0|e_0)$ to O.
3. O generates the licences UL_0, RdL_0 then computes $K_0 = D_{pub}(K'|d_0)$, $r_0 = Sig(H(H(X)||l_X)|d_0)$ and sends $(r_0, Y = E_{sym}(UL_0, RdL_0, n_0, H(X), l_X|K_0))$ to L.
4. L finds $D_{sym}(Y|K_0) = (UL_0, RdL_0, n_0, H(X), l_X)$, verifies r_0, UL_0 and Rd_0. If they are correct, L adds to its database $(Cert_0, UL_0, RdL_0, n_0, r_0, H(X), l_X)$ and notifies O.
5. O generates a watermark signal W_{own} embeds into the content X to get X'. It then encrypts X' and uploads on its content server CS_0.

We now describe the interactive protocol and the computations performed by a distributor D_i with the license server L. D_i first downloads the content from the content server of the higher level distributor D_{i-1}. L establishes a session key K_i with D_i. All the further secure communication between them are performed with symmetric key cryptography using key K_i. D_i computes its watermark information r_i, its re-distribution license RdL_i and usage license UL_i and sends to L after encryption with K_i. L decrypts the encrypted licenses using K_i and verifies the licenses and r_i. If all the verifications pass through, L clears D_i to upload the encrypted content on its content server CS_i. Formally the steps are as follows.

1. D_i downloads content from the content server CS_{i-1} and sends the request for the redistribution licence of D_{i-1} and $Cert_i$ to L.
2. L verifies $Cert_i$, extracts e_i, generates a random session key K_i, computes $K' = E_{pub}(K_i|e_i)$ and sends $(K', Y = E_{sym}(RdL_{i-1}, H(X), l_X|K_i))$ to D_i.
3. D_i generates the licences UL_i and RdL_i, computes $K_i = D_{pub}(K'|d_i)$, $D_{sym}(Y|K_i) = (RdL_{i-1}, H(X), l_X)$, $r_i = Sig(H(H(X)||l_X)|d_i)$, encrypts them to get $Y = E_{sym}(UL_i, RdL_i, n_i|K_i)$ and sends (Y, r_i) to L.
4. L computes $D_{sym}(Y|K_i) = (UL_i, RdL_i, n_i)$, verifies r_i, UL_i and RdL_i. If everything is correct, it adds to its database $(Cert_i, UL_i, RdL_i, r_i, n_i)$ and notifies D_i.
5. D_i uploads (encrypted) content on content server CS_i.

In the final stage, a consumer C (DRM agent) downloads the content (in a privacy preserving manner as described in Section 2.6) from the content server CS_k of the distributor D_k. It then proceeds for an interactive protocol with the license server L. L establishes a session key K_{k+1} for secure communication between them. C computes its watermark information r_{k+1} and generates a random prime number n_{k+1} (to preserve anonymity) different from $n_0, \ldots, n_k$, for the joint watermark computation and sends them to L. L after verification of r_{k+1} and n_{k+1} generates the joint watermark information of all the entities involved using CRT. L then passes the joint watermark information I to C. C then generates the watermark signal W from I and then embeds into the content X using a standard robust watermarking algorithm. Formally the steps are as follows.

1. DRM agent downloads the content anonymously (with privacy) from content server CS_k of the distributor D_k, sends $Cert_{k+1}$ to L and requests for starting a session.
2. L verifies $Cert_{k+1}$, extracts e_{k+1}, generates a random session key K_{k+1} encrypts as $K' = E_{pub}(K_{k+1}|e_{k+1})$ and sends K' to C.
3. DRM agent computes $K_{k+1} = D_{pub}(K'|d_{k+1})$ and sends to L the request for the usage licence of the distributor after encrypting both the identity of D_k and the identifier for the content X with K_{k+1}.
4. L decrypts the identity of D_k and the identifier for the content X with K_{k+1}, searches the database, computes $Y = E_{sym}(UL_k, H(X), l_X|K_{k+1})$ and sends Y to C.
5. DRM agent generates a random prime number n_{k+1}, computes $D_{sym}(Y|K_{k+1}) = (UL_k, H(X), l_X)$, $r_{k+1} = Sig(H(H(X)||l_X)|d_{k+1})$, $SIG(n_{k+1}) = Sig(n_{k+1}|d_{k+1})$, $Y = E_{sym}(n_{k+1}|K_{k+1})$ and sends $(r_{k+1}, Y, SIG(n_{k+1}))$ to L.
6. L computes $D_{sym}(Y|K_{k+1}) = n_{k+1}$ and verifies r_{k+1} and $SIG(n_{k+1})$. If all verifications pass through, it

adds the entry $(Cert_{k+1}, n_{k+1}, r_{k+1}, SIG(n_{k+1}))$ to its database. It then computes I using Equation 2 and sends $Y = E_{sym}(UL_0, I|K_{k+1})$ to C.

7. DRM agent computes $D_{sym}(Y|K_{k+1}) = (UL_0, I)$ and opens the content using the keys in UL_0 and UL_k to get X'. It computes watermark signal W from I using $W_{gen}(.)$ and then embeds into the content X' using $W_{emb}(.|K_X)$ (the key K_X depends only on the content and is common to all the consumers using X).

REMARK 3.1. *In Step 5 of the above protocol, n_{k+1} generated by the DRM agent needs to be different from $n_0, \ldots, n_k$ as the n_i's occurring in Equation 2 are to be relatively prime to each other. If $\pi(x)$ denotes the number of prime numbers less than or equal to x, then from the prime number theorem, $\lim_{x\to\infty} \pi(x) = \frac{x}{\ln x}$. Therefore, the number of t bit primes is approximately, $\frac{2^t-1}{t} - \frac{2^{t-1}-1}{t-1} \approx \frac{2^{t-1}}{t}$. Hence the probability that the choice of n_{k+1} is different from $n_0, \ldots, n_k$ is approximately $\frac{\frac{2^{t-1}}{t} - (k+1)}{\frac{2^{t-1}}{t}} \approx 1$, since $k \ll t$ (k is in the order of unity and t is in the order of hundreds). Thus n_{k+1} will be different from $n_0, \ldots, n_k$ with very high probability.*

3.4 The Watermarking Detection and Traitor Tracing Protocol

Suppose that the owner O found an illegal copy Y of the content X. Let J denotes a judge for arbitration. Then the watermark detection and traitor tracing protocol is as follows:

1. O checks whether its watermark signal W_{own} is present in Y. If so, O presents $(Y, W_{own}, H(X), l_X)$ to J.

2. J checks whether W_{own} is present in Y. If it is not present J ends the protocol, else proceeds.

3. J gets the watermark information I from the license server L and computes the watermark signal $W = W_{gen}(I)$. It obtains K'_X (publicly available) and checks whether the watermark signal W is present in Y using the detection algorithm $W_{det}(.|K'_X)$. If W cannot be detected in Y, J ends the protocol, else proceeds.

4. J gets $(n_{k+1}, SIG(n_{k+1}), Cert_{k+1})$ from L.

5. J computes r_{k+1} from $I \equiv r_{k+1} \mod n_{k+1}$.

6. J checks whether r_{k+1} is a valid watermark information of the consumer C by verifying whether r_{k+1} is a valid signature of C and n_{k+1} is a prime number generated by C by verifying the signature $SIG(n_{k+1})$. If both verifications pass through, J concludes that C was the consumer and hence was the traitor.

In the following section, we show that the security concerns listed in Section 2.3 are satisfied by our protocol.

3.5 Security Analysis

The soundness and completeness of the protocol rely on the security and robustness of the underlying cryptographic and watermarking primitives and the trustworthiness of the license server and the DRM agent.

If the owner finds an illegal copy of the content, the traitors can be identified using the protocol given in Section 3.4. Further, the scheme offers protection for parties who were not associated with the content against wrong identification or false framing as follows. Let n and (e, d) be the parameters of a party. The judge J, computes r from the equation $I \equiv r \mod n$, and checks whether r is a valid signature of that party. However, if that party was not involved, this verification will fail as its success corresponds to the *existential forgery* of the signature $Sig(H(H(X)||l_X)|d)$, which is not possible as the underlying digital signature scheme is assumed to be secure.

The individual watermark information r_i and hence the joint watermark information I is generated as a function of the content $(H(X))$ as well as the copy identifier (l_X). Thus, the watermark signal W is bound to the content and since it is generated and embedded by the DRM agent, the owner or distributors cannot create copies of the original content containing the consumer's watermark. Thus a malicious owner or distributor cannot frame an innocent sub-distributor or consumer by embedding the sub-distributor's or consumer's watermark into another content and accuse them of illegal distribution of the content.

Collusion and replay attacks are not possible, since the license server verifies r_i and stores them in the database. The redistribution license of D_i is accepted by the license server only if r_i was correctly generated. In the final stage, license server verifies the watermark information r_{k+1} of the consumer and generates the joint watermark information I. The watermark signal W is generated from I and embedded into the content by the DRM agent. DRM agent is the owner's entity residing in a consumer's machine and performs actions on contents according to the usage licenses. Since DRM agent is a trusted entity representing the owner, these steps will be carried out correctly. If not, the owner will not be able to trace the traitors and the distributors if he finds illegal copies in the future. Thus the watermark signal will be correctly embedded into the content.

The scheme protects the privacy concerns of a consumer. As described in Section 2.6, the consumer can download the content while maintaining privacy. Further, I does not reveal the identity of the consumer as n_{k+1} is not public. While interacting with the license server, consumer maintains privacy by sending only encrypted information about the content it downloaded. The watermark signal embedding key K_X and the detection key K'_X depends only on the content and is common for all the consumers using the content X. This choice also ensures privacy to the consumers.

3.6 Complexity of the Proposed Scheme

Most of the encryption operations performed are symmetric-key cryptography based to minimize the costly public-key cryptographic operations. Assume that there are k distributors. The summary of the computations performed by the owner, distributors, consumer and the license server are listed in Table 1. (+1)/(+2) appearing in the table denotes the encryption/decryption on the digital content.

The owner, needs to store its watermark signal W_{own} privately. Owner and the distributors need to store the encrypted content on their content servers. The consumer needs to store the parameter n_{k+1}, the downloaded content and the licenses. The license server needs to store $k+2$ digital certificates, $k+1$ usage and redistribution licenses each, $k+3$ digital signatures, $k+2$ prime numbers, hash of the content $h(X)$, identifier for that content l_X and the joint watermark information I.

Table 1: Complexity

	Owner	Distributor	Consumer	L
Communication	2	2	3	2k+5
Sym-key Enc	5 (+2)	3 (+1)	3	3k+5
Sym-key Dec	0	3 (+1)	5 (+2)	3k+8
Pub-key Enc	0	0	0	k+2
Pub-key Dec	1	1	1	0
Sign Generation	1	1	2	0
Sign Verification	0	0	0	2k+5

3.7 System Implementation

A general purpose CPU is embedded in owner's device. Owner's device has four components: a main controller, a computing module, a secure storage device and a content storage device. Upon creation of a content the main controller generates the licenses and contacts the license server. It stores its watermark information r_0, watermark w_{own} and the keys in the secure storage device. The main controller directs the computing module, to perform license generation, cryptographic operations, watermark generation/insertion operations and upload the content on the content server. A distributor's device and the roles of the components are similar to the case of the owner. In addition to that it should have a mechanism to detect the violations of the redistribution licence stored in the distributor's device.

The computational requirements of the consumer can also be taken care with a general purpose CPU or a special purpose processor in its device. A DRM agent is installed in the consumer's device. It is necessary to separate the consumer from the DRM agent, and place plug-ins between them. DRM agent may be viewed as consisting of three components: main controller, computing module and a secure storage device. Usage licenses, cryptographic keys and other user parameters are stored in the secure storage device. When a user tries to play a downloaded content for the first time, the main controller contacts the license server and gets the licenses and watermark information. It directs the computing module to decrypt the content and then generate and embed the watermark signal. The content is then re-encrypted and stored in the hard disk of the device as well as made available to the consumer to use. Whenever the consumer wants to reuse the content, the main controller searches for the corresponding license from secure storage device. If a license exists with a valid permission, it directs the computing module to decrypt the content and makes it available for the user.

The license server needs network connectivity to check the certificate status of the owner, the distributors and the consumers for authentication purpose. We can make use of PKI such as X.509 for this. Since the cryptographic load on the license server is more, it may be provided with cryptographic hardware accelerators in order to operate efficiently.

4. CONCLUSION

In this paper, we presented a mechanism for securely delivering multimedia content through a multiparty multilevel DRM architecture using a joint watermarking mechanism combined with cryptographic mechanisms. The protocol takes care of the security concerns of all the parties and only two watermark signals are embedded into the content compared to the embedding of multiple watermark signals into the content with the naive approach. Thus, this approach minimizes the possible degradation of the quality of a digital content due to embedding of watermark signals. Further, in case the owner or a distributor finds an unauthorized copy, they can identify the traitor with the help of a judge.

As a future direction of research, the protocols may be improved to reduce the dependence and the computational load on the license server and the individual watermark information may be computed as any other easily verifiable watermark information than digital signatures.

Acknowledgments

We thank the anonymous reviewers for their helpful comments on this paper. This work was supported by the Agency for Science, Technology and Research (A*STAR), Singapore under the project 'Digital Rights Violation Detection for Digital Asset Management' (Project No: 0721010022).

5. REFERENCES

[1] S. Chen, S. Chen, H. Guo, B. Shen, S. Jajodia, "Efficient Proxy-Based Internet Media Distribution Control and Privacy Protection Infrastructure", Quality of Service, IWQoS 2006, pp. 209-218, 2006.

[2] B. Hu, W. Ye, Sui-Li Feng, Xiao-Liang Wang, X. Xie, "Key Distribution Scheme Based on Two Cryptosystems for Hierarchical Access Control", ICACT 2006, pp. 1723-1728, Feb 20-22, 2006.

[3] S. O. Hwang, K. S. Yoon, K. P. Jun, K. H. Lee, "Modeling and Implementation of Digital Rights", J. of Systems and Software, V. 73, Iss. 3, pp 533-549, 2004.

[4] H. S. Ju, H. J. Kim, D. H. Lee, J. I. Lim. "An Anonymous Buyer-Seller Watermarking Protocol with Anonymity Control", Information Security and Cryptology, ICISC 2002", pp. 421-432, 2003.

[5] W. Kanjanarin, T. Amornraksa, "Scrambling and Key Distribution Scheme for Digital Television", ICON, 01.

[6] X. Liu, T. Huang, and L. Huo, "A DRM Architecture for Manageable P2P Based IPTV System", ICME 07.

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

[8] N. Memon, P. W. Wong, "A Buyer Seller Watermarking Protocol", IEEE Transactions on Image Processing, Vol. 10, No. 4, pp.643-649, 2001.

[9] A. Menezes, P. v. Oorschot, S. Vanstone, *Handbook of Applied Cryptography*, CRC-Press, pp. 610-612, 1996.

[10] H. Muratani, "Optimization and Evaluation of Randomized c-Secure CRT Code Defined on Polynomial Ring", IH 2004, pp. 282-292, 2004.

[11] 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.

[12] A. Sachan, S. Emmanuel, A. Das, M. Kankanhalli, "Privacy Preserving Multiparty Multilevel DRM Architecture", Workshop on Digital Rights Management, CCNC 2009, Jan.10-13, 2009.

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