Joint Watermarking Scheme for Multiparty Multilevel DRM Architecture

Tony Thomas, Sabu Emmanuel, *Member, IEEE* , A. V. Subramanyam,Mohan S. Kankanhalli,*Member, IEEE*

Abstract—Multiparty multilevel digital rights management (DRM) architecture involving several levels of distributors in between an owner and a consumer has been suggested as an alternative business model to the traditional two-party (buyer-seller) DRM architecture for digital content delivery. In the two-party DRM architecture, cryptographic techniques are used for secure delivery of the content and watermarking techniques are used for protecting the rights of the seller and the buyer. The cryptographic protocols used in two-party case for secure content delivery can be directly applied to the multiparty multilevel case. However, the watermarking protocols used in two-party case may not directly carry over to the multiparty multilevel case as it need to address the simultaneous security concerns of multiple parties such as owner, multiple levels of distributors and consumers. Towards this, in this paper we propose a joint digital watermarking scheme using Chinese remainder theorem for the multiparty multilevel DRM architecture. In the proposed scheme, a watermark information is jointly created by all the parties involved and then a watermark signal is generated out of it and embedded into the content. This scheme takes care of the security concerns of all the parties involved. Further, in the event of finding an illegal copy of the content, the traitors can be traced back.

Index Terms—digital rights management, watermarking, Chinese remainder theorem

1 Introduction

The ease with which digital contents can be obtained, replicated and distributed without any loss of quality has resulted in the widespread illegal replications and distributions of digital contents. Hence to prevent this and protect the intellectual property rights, DRM (Digital Rights Management) technologies have been developed. DRM uses cryptographic 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. In DRM, encryption is used to prevent unauthorized access to a content and watermarking is used to establish and prove ownership rights and to trace copyright violators by embedding the seller's and buyer's information into the digital content.

The traditional two party digital rights management (DRM) 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 traditional two party (buyer-seller) DRM architecture by many authors [12], [22]. The term multiparty refers to the multiple parties such as the owner, distributors, sub-distributors and consumers and the term multilevel refers to the multiple levels of distributors/sub-distributors involved in the distribution chain of a content.

In a multiparty multilevel DRM architecture, 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 MPML-DRM-A using Chinese remainder theorem (CRT) [20]. Our approach is to embed into the content only one watermark signal generated from a watermark information jointly generated by all the parties involved. We generate this joint watermark information using CRT. This was motivated by an application of CRT in secure broadcasting, effected by means of a secure lock by Chiou et al. [4]. The authors implemented this lock by using CRT. Analogously, we lock the identities of all the entities using CRT as a watermark and is embedded into the content. Our scheme thus takes care of the security concerns of all the parties involved. Further, in the event of finding an illegal copy of the content, the traitors can be traced.

The rest of the paper is organized as follows. In Section 2, we discuss the preliminaries required. Section 3, includes our joint watermarking mechanism, its security and complexity analysis. A discussion on the implementation of the proposed scheme is given in Section 4. In Section 5, we do a comparison of the proposed approach with the extensions of two-party solutions.The paper concludes with some remarks and future directions for

- *T.Thomas, S. Emmanuel and A. V. Subramanyam are with the School of Computer Engineering, Nanyang Technological University, Singapore. E-mail: {ttony,asemmanuel,subr0021}@ntu.edu.sg*
- *M. S. Kankanhalli is with School of Computing, National University of Singapore, Singapore.E-mail: mohan@comp.nus.edu.sg*

research in Section 6. A preliminary version of the paper appeared in NOSSDAV 2009 [23].

2 PRELIMINARIES

In this section, we briefly discuss the preliminaries required for this paper.

2.1 Multiparty Multilevel DRM Architecture

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 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 locations. Many of the current practices such as Apple i-tunes lack this much flexibility. Apples i-Tunes music store lets customers search a catalogue of tracks. With one click, users can purchase the songs and download them. i-tunes uses Apple's FairPlay digital rights management (DRM) system which limits and controls its usage. To the best of our knowledge there is no such mechanism like a joint watermarking or traitor tracing mechanism in i-tune and is using only copy-protection software with cryptographic mechanisms to stick to certain devices on which it can play. Hence we adopt a multiparty multilevel DRM architecture (MPML-DRM-A) given in Fig.1 as our content distribution model. The owner and distributors maintain their own content servers CS. To ensure the security of the content, the content is stored in encrypted manner on the content servers. The license server issues redistribution licenses to distributors and usage licence to the consumers. A license grants the receiver specific permissions, constraints and content decryption keys. A consumer is allowed to get the content from any of the content servers. We intend to build a joint watermarking mechanism into this architecture to take care of all the copyright issues.

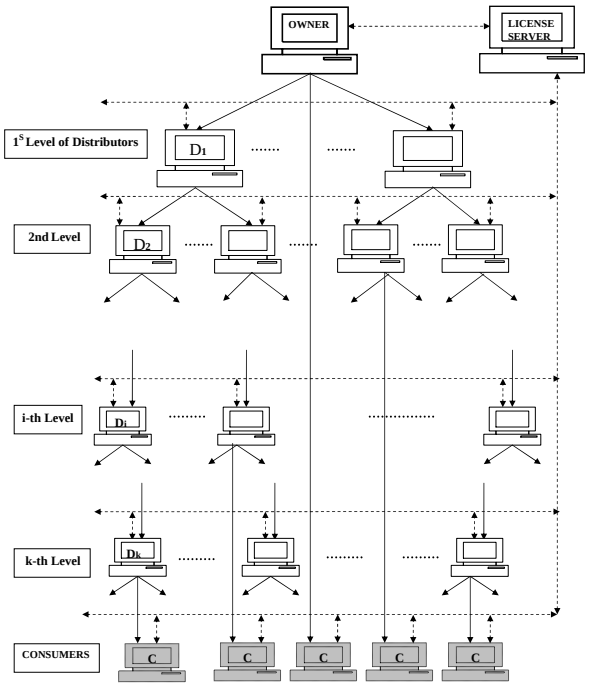

Fig. 1. MPML-DRM-A: The dark arrows show the flow of the content and dotted arrows show the communication between an entity and the license server.

2.2 Structure of Licenses

Licenses are created by the owner and distributors for other distributors and consumers. License contains the following entries: identity of the license issuing party, identity of the content(s), permissions, constraints and keys required for taking appropriate action. There are two types of licenses in this architecture: *redistribution license* (RL) and *usage license* (UL).

Redistribution licenses are created by the owner or a distributor for another distributor lower in the distribution chain. The redistribution licenses contain secret keys of the party which generates license for a particular content. Permissions include permission for content redistribution and permission to issue redistribution licenses. Constraints associated with permissions can be time based, count based, and location based. Enforcement of redistribution license is done with the help of a license server, which keeps record of redistribution and usage licenses issued by owner/distributor.

Usage licenses are created by the owners and distributors for the consumers to use the content. Consumers need to get the usage license of the owner as well as that of the distributor from whom the content was downloaded. Usage licenses contain the keys for opening the content and the permissions and constraints for using the content. Enforcement of usage license is done with the help of a trusted DRM agent at consumers machine.

2.3 Related Works and CRT

There exists several joint watermarking mechanisms [19], [15], [8] for the two party (buyer-seller) DRM architecture. However, multiparty multilevel DRM architecture being a recent business model, there has not been any work on joint watermarking for this architecture yet. Our Chinese remainder theorem(CRT) based joint watermarking scheme seems to be the first in this direction. CRT is as follows. Let $n_1, \ldots, n_k$ are pairwise coprime positive integers and $r_1, \ldots, r_k$ are any collection of integers. Then the k congruences

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

has a unique solution x such that $0 \leq x < N = n_1 \dots n_k$.

CRT has been used in several secure broadcast communication and DRM applications. Some such applications are a secure broadcast mechanism by Chiou and Chen [4], a key distribution scheme using CRT for conditional access system in digital TV broadcast [11], a CRT based parameter distribution in the scrambling process for conditional access to Pay-TV systems [16] and a binary fingerprinting code using CRT [21].

2.4 Notations

We follow the following notations throughout this paper.

- The entities involved are: an owner O, k distributors $D_1, \dots, D_k$, a consumer C and a license server L.
- X denotes the content, l_X be a unique copy number of the content X.
- $E_{pub}(.|K)$, $D_{pub}(.|K)$, $Sig(.|K)$ and $Ver(.|K)$ denote the encryption, decryption, digital signature generation and digital signature verification algorithms (with key K) respectively, corresponding to a standard public key cryptosystem.
- For $0 \leq i \leq k+1$, (e_i, d_i) and $Cert_i$ denote the public-private-key pairs and the public-key certificate assigned to the owner, the k distributors and the consumer respectively.
- For $i = 0, \dots, k$, CS_i denotes the content server of the owner and the k distributors $D_1, \dots, D_k$ respectively.
- $E_{sym}(.|K)$, $D_{sym}(.|K)$ denote the encryption and decryption algorithms corresponding to a standard symmetric-key cryptosystem like AES or 3DES.
- $E_{sym}(x_1, ., x_p|K) = (E_{sym}(x_1|K), \dots, E_{sym}(x_p|K))$, $D_{sym}(y_1, .., y_p|K) = (D_{sym}(y_1|K), \dots, D_{sym}(y_p|K))$.
- $H(.)$ denotes any standard hash function such as SHA1 or MD5.
- $PRNG(.)$ denotes a binary pseudo random number generator.
- Let $||$ denotes the concatenation operator, for $0 \leq i \leq k+1$ let $Y = H(H(X)||l_X)$ and $r_i = Sig(Y|d_i)$.
- $W_{gen}(.|K)$ denotes any standard watermark signal generation algorithm (with key K) from watermark information.
- $W_{emb}(.|K)$ denotes any standard robust watermark embedding algorithm and $W_{det}(.|K)$ denotes the corresponding watermark detection algorithm (with key K).
- K_X denotes the key used for embedding the watermark signal in the content X and K'_X denotes the key used for detecting the same watermark signal.
- I denotes the joint watermark information and W denotes the joint watermark signal.
- J denotes a judge who is called for arbitration in case of a dispute.
- For $i = 0, \dots, k$, UL_i and RdL_i denotes the usage and redistribution licenses of O, D_1,…,D_k respectively.

3 The Proposed Watermarking Scheme

In this section, we describe our joint watermarking protocol based on CRT for MPML-DRM-A. The proposed watermarking protocol involves the following entities: an owner O, k levels of distributors $D_1, \dots, D_k$ (there can be no distributor also), 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.

3.1 Generation of Individual Watermark Information

Each party i (owner or distributor or consumer) involved in the content delivery generates its individual watermark information r_i using its private key d_i as its digital signature $r_i = Sig(Y|d_i)$, where $Y = H(H(X)||l_X)$.

3.2 Generation of Joint Watermark Information

Let $r_0, r_1, \dots, r_k$ and r_{k+1} be the individual watermark information of the parties $O, D_1, \dots, D_k$ and C respectively, computed as digital signatures as described in the previous section. Let $n_0, n_1, \dots, n_{k+1}$ be relatively prime integers assigned to these parties respectively. Then their joint watermark information I is the solution of the following set of $k+2$ congruences:

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

The existence and uniqueness of I is guaranteed by CRT.

3.3 Joint Watermarking Protocol

Recall the notations given in Section 2.4. Let a content reaches a consumer C from the owner O through k distributors $D_1, \dots, D_k$. We now describe the watermarking protocol below. The steps performed by the owner, distributors and the consumer are separately described.

We begin with the interactive protocol and the computations performed by the owner with the license server.

1) O sends, $Cert_0$ to the Licence server L.
2) L verifies $Cert_0$, extracts the public key e_0 of O from $Cert_0$, generates a random session key K_0 and sends $K' = E_{pub}(K_0|e_0)$ to O.
3) O decrypts, $K_0 = D_{pub}(K'|d_0)$, computes its watermark information $r_0 = Sig(H(H(X)||l_X)|d_0)$ and generates its usage license UL_0 and redistribution licence RdL_0. It encrypts them using K_0 as, $Y = E_{sym}(r_0, UL_0, RdL_0, n_0, H(X), l_X|K_0)$.
4) O sends, Y to L.
5) L does, $D_{sym}(Y|K_0) = (r_0, UL_0, RdL_0, n_0, H(X), l_X)$. It verifies r_0 and checks the licenses 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.

6) O generates a unique watermark signal W_{own} as a function of l_X and 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 L.

1) D_i downloads (encrypted) content from the content server CS_{i-1} and sends the request for the redistribution licence of D_{i-1} along with its public-key certificate $Cert_i$ to L.
2) L verifies the public-key certificate $Cert_i$ and extracts the public key e_i of D_i from $Cert_i$. It generates a random session key K_i, encrypts it using the public-key of D_i as $K' = E_{pub}(K_i|e_i)$ and then encrypts the licence and other parameters using the session key as, $Y = E_{sym}(RdL_{i-1}, H(X), l_X|K_i)$ and sends (Y, K') to D_i.
3) D_i decrypts K' and Y as $K_i = D_{pub}(K'|d_i)$ and $D_{sym}(Y|K_i) = (RdL_{i-1}, H(X), l_X)$. It computes its watermark information $r_i = Sig(H(H(X)||l_X)|d_i)$ and generates its usage licence UL_i and redistribution licence RdL_i and encrypts to get $Y = E_{sym}(r_i, UL_i, RdL_i, n_i|K_i)$ and sends Y to L.
4) L decrypts Y as, $D_{sym}(Y|K_i) = (r_i, UL_i, RdL_i, n_i)$. It then verifies r_i, UL_i and RdL_i. If they are correctly generated, 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 its content server CS_i.

In the final stage, a consumer C downloads the content from the content server of the distributor D_k. The consumer's machine has a separate module for storing its secret keys and carrying out sensitive/secure computations. In the following protocol, **C** denotes this module. Further, the consumer has installed a DRM agent of the owner in its machine. Although DRM agent is an entity of the owner it is hard for it to collude with the owner against the distributors and consumers. The DRM agent is provided only with the joint watermark information and the options in front of the DRM agent are to embed the joint watermark or not and the DRM agent will do the embedding for the sake of the owner. Further, since it does not have access to the watermark information of the other entities (including that of the consumer), it can not act against them. We assume that there is cryptographic key K_{drm} associated with the DRM agent. The license server can find this key in the usage license UL_0 of the owner. Formally the steps are as follows.

1) **C** downloads (encrypted) content anonymously from the content server CS_k of the distributor D_k and sends the public-key certificate $Cert_{k+1}$ to L and requests for starting a session.
2) L verifies $Cert_{k+1}$, extracts the public key e_{k+1} from $Cert_{k+1}$, generates a random session key K_{k+1} encrypts using the public-key of C as $K' = E_{pub}(K_{k+1}|e_{k+1})$ and sends K' to C.
3) **C** decrypts K' as $K_{k+1} = D_{pub}(K'|d_{k+1})$ and then 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} and then identifies the content information, the owner and all the distributors associated with it in its database. It then encrypts the usage licence of the distributor and other parameters to get $Y_1 = E_{sym}(H(X), l_X|K_{k+1})$, $Y_2 = E_{sym}(UL_k|K_{drm})$ and sends (Y_1, Y_2) to C.
5) **C** decrypts Y_1 as $D_{sym}(Y_1|K_{k+1}) = (H(X), l_X)$ and DRM agent decrypts Y_2 as $D_{sym}(Y_2|K_{drm}) = UL_k$.
6) **C** computes the watermark information of the consumer as, $r_{k+1} = Sig(H(H(X)||l_X)|d_{k+1})$, generates a random number n_{k+1} coprime to $n_0, \ldots, n_k$, then digitally signs it as $SIG(n_{k+1}) = Sig(n_{k+1}|d_{k+1})$, encrypts r_{k+1} and n_{k+1} as $Y = E_{sym}(r_{k+1}, n_{k+1}|K_{k+1})$ and sends $(Y, SIG(n_{k+1}))$ to L.
7) L computes $D_{sym}(Y|K_{k+1}) = (r_{k+1}, n_{k+1})$ and checks that for all $0 \leq i \leq k$, n_{k+1} is coprime to n_i. If it does not hold L requests C to resend n_{k+1} and $SIG(n_{k+1})$. It then verifies signatures r_{k+1} and $SIG(n_{k+1})$. If all verifications pass through, it adds to its database the entry $(Cert_{k+1}, n_{k+1}, r_{k+1}, SIG(n_{k+1}))$.
8) L computes the joint watermark information I as the CRT solution of the following equations,

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

encrypts and sends $Y = E_{sym}(UL_0, I|K_{drm})$ to C.
9) DRM agent decrypts, $D_{sym}(Y|K_{drm}) = (UL_0, I)$, opens the content using the keys in UL_0 and UL_k to get X', computes a watermark signal W from the watermark information I using the watermark signal generation algorithm $W_{gen}(.)$ and then embeds into the content X' using the watermark signal embedding algorithm $W_{emb}(.|K_X)$.

3.4 Watermarking Detection and Traitor Tracing

We assumed that the watermark signal W is generated and then embedded using a well known robust watermarking algorithm. Suppose that the owner O found an illegal copy Y of his content X. Let J denotes a judge for arbitration. The traitor tracing protocol is as follows:

1) O checks whether its watermark signal W_{own} is present in the content Y. If it is not present END the protocol, else proceed.
2) O presents $(Y, W_{own}, H(X), l_X)$ to J.
3) J checks whether W_{own} is present in Y. If it is not present END the protocol, else proceed.
4) J gets the joint watermark information I from the license server L and computes the watermark signal $W = W_{gen}(I)$. It obtains K'_X and checks whether W is present in Y using the detection algorithm $W_{det}(.|K'_X)$. If W cannot be detected in Y, END the protocol, else proceed.

TABLE 1
Computational Complexity

	O	D_i	C	L
Symmetric-key Encryption	6	4	4	3k+5
Symmetric-key Decryption	0	3	5	4k+10
Public-key Encryption	0	0	0	k+2
Public-key Decryption	1	1	1	0
Digital Signature Generation	1	1	2	0
Digital Signature Verification	0	0	0	2k+5

5) J gets $(n_{k+1}, SIG(n_{k+1}), Cert_{k+1})$ from L.
6) J computes r_{k+1} from the equation $I \equiv r_{k+1} \mod n_{k+1}$.
7) 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 random number generated by C by verifying the signature $SIG(n_{k+1})$. If both verifications pass through, J concludes that C was the consumer associated with that content and hence was the traitor.

Note that, in the proposed solution the distributors are not able to identify an illegal copy distributed without the help of the license server. Only the owner has this possibility, thanks to the unique watermark W_{own} of the owner embedded into the content by the owner.

3.5 Complexity of the Protocol

In this section, we analyze the complexity of the proposed scheme. Most of the encryption operations used are symmetric-key cryptography based to minimize the costly public-key cryptographic operations. The public-key certificates $Cert_o, \ldots, Cert_{k+1}$, the parameters $n_0, \ldots, n_k$ in the CRT equations and the watermark detection key K'_X are publicly available. We now analyze the communication, computational and storage complexity of the proposed scheme.

Assume that there are k distributors. The license server L, exchanges a total of $2k + 5$ messages. L needs to verify $2k+5$ digital signatures, compute $k+2$ public key encryption, $3k+5$ symmetric-key encryptions and $3k+8$ symmetric-key decryptions. It also generates $k + 2$ random numbers, performs one Chinese remainder theorem computation (if n_i's are t bit numbers the complexity for this computations $O(kt^2)$ [20]) and verifies $k + 1$ usage licenses and redistributions licenses each. 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. Thus the over all complexity of L is linear in the number of distributors k. Practically k is a small number. The complexity extends linearly when L is serving multiple consumers.

The major computations of owner, distributors, consumer and license server are summarized in Table 1.

3.6 Security Analysis

We will now do the security analysis of our protocol. 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.

1) **Traitor Tracing**: If the owner finds an illegal copy of the content, he can identify all the distributors and the consumer involved in the distribution of the content using the protocol given in Section 3.4.
2) **Security Against False Framing**: The scheme offers protection for parties who were not associated with a content against wrong identification or false framing as follows. Let n and (e, d) be the parameters of a party. The judge computes r from the equation $I \equiv r \mod n$, and checks whether r is a valid signature of that party by checking whether $Ver(r, H(H(X)||l_X)|e) = 1$ holds. If the 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 secure.
3) **Rights of Consumer and Distributors**: Since watermark signal is generated and then embedded by the DRM agent, the owners or distributors cannot create copies of the original content containing the consumer's watermark. Further, since the watermark signal is formed from the joint watermark information, the owner or distributor will not be able to frame false allegations against a lower level sub-distributor or consumer regarding illegal distribution of a content.
4) **Binding of Watermark to Content**: The individual watermark information r_i and hence the joint watermark information I are generated as a function of the content ($H(X)$) and the identifier (l_X). Thus, the watermark signal W is bound to the content.
5) **Proof of Ownership and Distributorship**: In case of a dispute, the owner or distributor can settle the dispute using the protocol given in Section 3.4 with the help of a judge.
6) **Collusion Attack**: The term 'collusion attack' in the watermarking literature usually refers to a coalition of users that compare their watermarked contents in order to gain information about the watermarking process and/or remove the watermark. These type of attacks are not specific to our proposal and depends on the strength and robustness of the specific watermarking algorithm used. In our case, the individual watermark information r_i are generated by the parties themselves as their digital signature. The license server verifies r_i and stores them in the database. This prevents collusion attacks in the generation of I.
7) **Embedding of Correct Watermark Signal**: The license server verifies the individual watermark information r_i and stores them in the database. The

redistribution license of the i-th party is accepted by the license server only if r_i was correctly generated. In the final stage, the 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 then embedded into the content by the DRM agent. The DRM agent is the owner's entity residing in a consumer's machine and performing 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 if he finds illegal copies in the future as well as will not be able to trace the distributors involved in the distribution of his content. Thus the watermark signal will be correctly embedded into the content.

8) **Privacy/Anonymity of Consumers**: The scheme protects the privacy concerns of a consumer. The consumer downloads the content anonymously and generates a random number n_{k+1} towards generating the joint watermark information I. Thus I does not reveal the identity of the consumer. While interacting with the license server, the consumer maintains privacy by sending only encrypted information about the content it downloaded. Although, I does not reveal the identity of the consumer, if a need arises the consumer can be identified. Further, 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 same content X. This choice also ensures the privacy of the consumers.

4 Implementation of the Proposal

In this section, we discuss an implementation of the proposed scheme.

4.1 Secure Delivery of the Content

The owner and the distributors encrypt the content $\mathbf{X}$ before uploading on their content servers. The set of components of $\mathbf{X}$ to be encrypted is divided into two mutually disjoint sets of components $\mathbf{X_O}$ and $\mathbf{X_D}$. Now the owner (corresponding to role as owner) encrypts $\mathbf{X_O}$ and the distributors/owner (owner corresponding to role as a distributor) encrypt $\mathbf{X_D}$. For security reasons the set $\mathbf{X_O}$ should contain the major components of $\mathbf{X}$. The encryption of $\mathbf{X_O}$ and $\mathbf{X_D}$ are performed using symmetric key encryption algorithms. Whenever a distributor obtains the content from a higher level distributor/owner he decrypts the encryption (by that distributor/owner) of $\mathbf{X_D}$ using the key in the redistribution license and then re-encrypts $\mathbf{X_D}$ using his key and uploads the resultant content on his content server. A consumer downloads the content from the content server of any distributor/owner. It then obtains the usage licenses by the owner and the distributor from the license server. The usage licence of the owner contains the key for decrypting $\mathbf{X_O}$ and the usage licence of the distributor/owner contains the key for decrypting $\mathbf{X_D}$.

4.2 Joint Watermark Information Generation

All the parties get their cryptographic credentials (public-private keys and digital certificate) from a key generation/certifying authority. Further, the owners and distributors collect their CRT parameter n_i along with a certificate of its ownership from another or same key generation/certifying authority. SHA-1 [24] is chosen as the one-way hash function $H(.)$ for use.

The parties generate their individual watermark informations r_i as their elliptic curve digital signatures (ECDSA) [14] on $H(H(X)||l_X)$, where X is the content and the license server finally generates the joint watermark information as described in the Section 3.

For security level of 80 bits, the ECDSA signature size is 320 bits. The size t of the numbers n_i can be chosen as any number greater than 320. This is to ensure that r_i's do not get modulated out in the congruence relation $I \equiv r_i \mod n_i$, so that r_i's could be extracted out from I and verified. The authority generating n_i's need to ensure that they are coprime to each other (another way to generate n_i's is to allow O to generate a prime number n_0 of size t_0, each D_i to generate a prime number n_i of size t_i and C to generate a prime number n_{k+1} of size t_{k+1}individually such that $t_0 < t_1 < \cdots < t_k < t_{k+1}$). Since $0 \leq I \leq n_0 \ldots n_{k+1}$, the size of I is $(k+2)t$ bits. If we take $k = 8$ (8 distributors) and $t = 330$, the size of the watermark information becomes just 3300 bits. It is possible to bring down the size of I even further by using short signatures instead of ECDSA. For example, if we choose short signatures such as the one in [2], where the signature size is just 160 bits, by taking $k = 8$ and $t = 170$, the size of the watermark information can be bounded by just 1700 bits. Further, the discussions in the Section 4.3 will show that the size of I is not really a matter of concern.

4.3 Watermark Signal Generation

The DRM agent obtains the joint watermark information I from the licence server as described in the protocols in Section 3. Let $s = H(I)$ and p_0 be a seed obtained from s (p_0 could be a few bits of s). A pseudo random number sequence $p_1, p_2, \ldots$ is generated using $PRNG(.)$ with p_0 as seed. If the watermark signal needed to be embedded is not larger than 160 bits, it is advisable to generate $p_1, p_2, \ldots$ from s itself by interpreting 0 bits as -1 and 1 bits as 1, instead of using $PRNG(.)$ to prevent possible collisions in the generated watermark signal. The spread spectrum watermark signal $w_1, w_2, \ldots$ is generated from $p_1, p_2, \ldots$ as $w_i = \alpha_i.b_i.p_i$, for all $i \geq 1$, where $\alpha_i \geq 0$ is a locally adjustable amplitude factor and b_i is the spread sequence (see [9] for details). w_i is then embedded into the content using the spread spectrum watermarking techniques given in [9].

4.4 Watermark Signal Embedding

The watermark signal generation from I and embedding are carried out by the DRM agent in the consumer's device. This may me implemented using one-time pads [7] or stream switching [13] or joint decryption and watermarking [18] or lookup-table (LUT)-based ciphers [1], [3]. For simplicity, we assume that a tamper proof box or more specifically a Trusted Platform Module (TPM) is available at the consumers machine. Trusted Computing Group (TCG) has specified the components of a TPM which is a tamper resistant module and can be trusted to store security-sensitive data in ways testable by a remote party. TPM can enforce access control policies associated with a resource in such a way that a user cannot bypass these policies, whilst maintaining access to resource.

The watermark embedding key K_x is chosen as a function of the the media. For example, if $H(.)$ is a hash function and $X = \{x_1, x_2, \ldots, x_m\}$ is a representation of the media X as a vector of real components, then we may compute K_x as $K_x = H(y_1||y_2||\ldots||y_m)$ or a key derived from the above hash using a key derivation function, where $\forall i$, $y_i = |[x_i]|$ (absolute value of the integral part). We assume that this key is available in the usage license UL_0 of the owner.

4.5 Watermark Signal Detection

To detect the watermark information in a content, first the database of watermark informations are obtained from the license server. The pseudo random number sequences are then generated as described in the Section 4.3. The watermark signal is then detected in the content using the watermark detection algorithm given in [9]. The watermark detection key K'_X is either same as the embedding key K_X or if it is different it is obtained from the owner. The detection key K_X is available with the owner, license server (from UL_0) and the DRM agent.

5 Comparison of the Proposal

In this section, compare our approach with the extensions of buyer-seller protocol to the MPML-DRM-A.

5.1 Extensions of Buyer-Seller Protocols

In this section, we compare our approach with the extensions of buyer-seller watermarking protocols [19] and [17] to the multiparty multilevel architecture. The buyer-seller watermarking protocol is a three-party protocol between a seller, a buyer, and a trusted watermark certification/generation authority (WCA/WGA). The seller sells copies of the content to a buyer, logging all transactions in a local database, whose entries facilitate tracing of content. The role of WCA/WGA is to ensure an honest generation of watermarks and send them along with a time-stamp and a digital signature.

In [19] Memon et al. described a buyer-seller watermarking protocol using homomorphic public-key cryptosystems. Here the seller first embeds his watermark signal into the content and then embeds a transformation (permutation) of the watermark signal of the buyer into the already watermarked content and passes the resultant content to the buyer. The extension of this protocol to the MPML-DRM-A is as follows. The owner and D_1 executes a buyer seller watermarking protocol. D_1 gets the watermarked content (two watermark signals embedded). Then D_1 gets the watermark signal of D_2 (in encrypted format), embeds a transformation of it into the content and passes the resultant content to D_2. There are three watermark signals in the content now. It is easy to see that the security concerns of O, D_1 and D_2 are taken care here (D_1 need not have to put an additional watermark signal before passing the resultant content to D_2 as the role of seller's watermark signal is taken care by the joint watermark signal between O and D_1). Finally, the consumer passes his watermark signal (encrypted) to D_k and D_k embeds a transformation of it into the content received from D_{k-1}. D_k then passes the resultant content to the consumer C. We can see that there will be $k+2$ watermark signals in the content the consumer is receiving. In general, with this approach the number of watermark signals in the content increases linearly with the number of distributors and thus it is not scalable. With our joint watermarking mechanism, there will be only two watermark signals in the content which the consumer receives, irrespective of the number of distributors involved. This can be even reduced to just one watermark signal by embedding only the joint watermark into the content and ignoring the separate watermark signal (W_{own}) embedding by the owner. However, the main disadvantage of the scheme of Memon et al. is the use of costly (homomorphic) public-key encryption mechanisms. Further, the scheme will not work if an intermediate entity behaves maliciously. If any party embeds a wrong watermark, the whole watermark will get corrupted and it will not even possible to trace the malicious entity. So a trusted third party will be needed to ensure that each party performs its role correctly.

In [17] Katzenbeisser et al. proposed a buyer-seller watermarking protocol which avoids the use of homomorphic public-key encryption and uses a secure watermark embedding based on partial encryption. The two-party protocol of Katzenbeisser et al. may be described in simple terms as follows. Let WGA be a trusted watermark generation authority, X denotes a content (represented as a vector of quantized real numbers which denote samples in the spatial/temporal domain or coefficients in a transform domain), W denotes a watermark signal and K denotes an encryption key.

1) WGA generates W and K.
2) WGA sends K to O and $W \oplus K$ to C.
3) O computes $X \oplus K$ and sends to C.
4) C computes $(X \oplus K) \oplus (W \oplus K) = X \oplus W$.

Thus the consumer C gets the watermarked content $X \oplus W$. Note that the scheme works only if the mutual

cancellation in Step 4 holds. In particular, scheme does not work with any encryption of the content or any watermarking mechanism. A straight forward extension of the above protocol to our multi-party multi-level DRM architecture may be as follows: WGA choses random keys $K, K_1, \ldots, K_k$. WGA sends K to O, K_i to D_i, for $1 \leq i \leq k$ and $W \oplus K_c$ to C, where K_c is such that $K = K_1 \oplus \cdots \oplus K_k \oplus K_c$. O computes $X \oplus K$ and uploads on its content server. D_1 downloads $X \oplus K$, computes $X \oplus K \oplus K_1$ and uploads on its content server. Finally D_k computes $X \oplus K \oplus K_1 \oplus \cdots \oplus K_k$ and uploads on its content server. The consumer C downloads $X \oplus K \oplus K_1 \oplus \cdots \oplus K_k$ and obtains $W \oplus K_c$ from WGA. C computes $X \oplus K \oplus K_1 \oplus \cdots \oplus K_k \oplus W \oplus K_c = X \oplus W$. Thus the consumer C gets the watermarked content $X \oplus W$.

The main disadvantage with this approach is that it does not work with any encryption of the content or any watermarking of the content. Whereas, our approach is applicable to any encryption of the content and any watermarking of the content. The computational and communication load on the WGA under this scheme is comparable to that on the license server L in our scheme. WGA is required to perform many complex computations like: (PKC) decryption of session keys, generation of watermark sequence, generation of multiple keys for encrypting watermark, encryption of watermark (XOR), encryption of encrypted watermark (SKC), PKC encryption of watermarks and keys and digital signature generation [17]. The identity of all the entities involved can be immediately derived in a unique manner from the watermark in our case from the CRT equations, where as this is not immediate in this case as the key K could be splitted up in several ways which point to different entities at the same time. Further as in the previous case, the above extensions will succeed only if all the parties involved are honest. A trusted third party is needed to verify that each party is performing correct steps.

We conclude that the role license server L in our architecture is analogous to the role of WCA and WGA in the buyer-seller watermarking protocols and its extensions. Thus all these protocols use a trusted third party. Minimization of dependence on any of these trusted parties require implementation of more complex interactive protocols and cryptographic primitives.

5.2 Experimental Study

In this section, we carry out a set of experiments to evaluate the performance of our proposal. In the first set of experiments we study the effect of multiple watermarking on images. In the second set of experiments we fix the watermark strength and study the robustness of multiple watermarking on images.

5.2.1 Effect of Multiple Watermarking on Quality

In this section, we do a comparison of the naive extension of the buyer-seller watermarking (resulting in multiple watermarks on the content) with our approach.

Fig. 2. Lena: Original and Watermarked Images

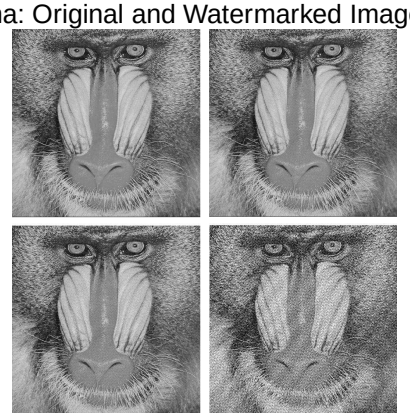

Fig. 3. Baboon: Original and Watermarked Images

Fig. 4. Pepper: Original and Watermarked Images

It is carried out using the spread spectrum watermarking algorithm of Hartung and Girod [9] with parameters $\alpha = 5$ as the amplification factor and $cr = 2400$ as the chip-rate. The results of the experiments with the standard test images of Lena, Baboon and Pepper are given in the Table 2 and illustrated in the figures Fig.2, Fig.3 and Fig.4. The four images in each figure are given in the following order: 1st row: original image and image with one watermark (left to right), 2nd row: image with two and ten watermarks (left to right) respectively. The figures clearly show the deterioration of the quality of the images with multiple watermarking. We expect similar results with other watermarking algorithms.

Peak signal to noise ratio ($PSNR$) and Structural

TABLE 2
PSNR and SSIM

	Single WM		Double WM		Multiple(10) WM	
Watermark	06.5234		12.5413		26.1225	
	PSNR	SSIM	PSNR	SSIM	PSNR	SSIM
Lena	37.1718	0.82156	31.1645	0.56598	17.9468	0.08309
Baboon	37.1701	0.93796	31.1495	0.81924	17.2268	0.27282
Pepper	37.2395	0.86246	31.2277	0.65612	17.5349	0.16106

SIMilarity (SSIM) index between the original and the watermarked images are given in Table 2.

$PSNR$ represents the ratio between the maximum possible power of a signal and the power of corrupting noise that affects the fidelity of its representation. Let $PV_{org}(i,j)$ and $PV_{wat}(i,j)$ denote the pixel value of the original image and the watermarked image at (i,j) ($m \times n$ gray-scale images). Then $PSNR$ is computed by the following equations:

$$PSNR = 20 \log \frac{255}{\sqrt{MSE}}, \text{ where}$$

$$MSE = \frac{1}{mn} \sum_{i=0}^{m-1} \sum_{j=0}^{n-1} (PV_{org}(i,j) - PV_{wat}(i,j))^2.$$

SSIM which is a perceptual measure is used for measuring the similarity between two images. SSIM is designed to improve on methods like PSNR and MSE [25]. The SSIM metric is calculated on various windows of an image. The measure between two windows of size $N \times N$, x and y is :

$$SSIM(x,y) = \frac{(2\mu_x\mu_y + c_1)(2cov_{xy} + c_2)}{(\mu_x^2 + \mu_y^2 + c_1)(\sigma_x^2 + \sigma_y^2 + c_2)},$$

where μ_x is the average of x, μ_y is the average of y, σ_x^2 is the variance of x, σ_y^2 is the variance of y, cov_{xy} is the covaraince of x and y, $c_1 = (k_1 L)^2$, $c_2 = (k_2 L)^2$, $L = 2^{\text{bits per pixel}} - 1$, $k_1 = 0.01$ and $k_2 = 0.03$.

SSIM index lies between -1 and 1, and value 1 is only reachable in the case of two identical sets of data. Typically it is calculated on window sizes of 8x8.

PSNR and SSIM of the images with single, double and multiple (10) watermarking is given in Table 2. The rapid decrease in the PSNR and SSIM along a row shows the deterioration of the quality of the content with multiple watermarking.

5.2.2 Robustness of Multiple Watermarking

In this section, we examine the probability of detection of the watermarks from a watermarked content. We computed the ratio of the total number of bits of a watermark correctly detected to the total number of bits in that watermark. We may interpret this ratio as the probability of detection of that watermark. Let p denotes the product of the ratios (of the number of bits detected to the total number of bits in the watermark) of all the watermarks embedded into the content. We may interpret p as the probability of detection of all the watermarks in the content. Let $n_{0.60}$ denotes the number of watermarks in a content where the ratio of number of bits detected to the total number (probability) is greater than 0.60 and $n_{0.75}$ denotes the number of watermarks where this ratio (probability) is greater than 0.75. We performed experiments with two watermarks and ten watermarks embedded on nine test images. The six additional test images used are given in Fig.5, which are Boat, Clown, Barbara, Goldhill, Airpalne and Fruit starting from top-left in the clockwise order. The watermarking algorithm and the chip-rate (cr) used are the same as that in the previous section. The parameter α and the number of bits in the watermark (W_{len}), are varied in each case to get different watermark strengths. The results of the experiments are given in Table 3.

Fig. 5. Boat, Clown, Barbara, Goldhill, Airplane and Fruit

We observed the following from our experiments.

1) If we fix W_{len} and increase α (this increases the watermark strength) the probability of detection of watermarks increases. This implies that higher probability of detection can be obtained at a cost on the quality of the content.
2) If we fix α and decrease W_{len} (this decreases watermark strength) the probability of detection of watermarks increases. This implies that higher probability of detection can be obtained at a cost on the security level of the watermark.

We conclude that with our approach the watermarks (two) can be detected without compromise on the quality or on the security, whereas with multiple watermarking, either the quality of the content or the security of the watermark is to be compromised to detect all the watermarks embedded into the content.

6 Conclusion

In this paper, we presented a joint watermark protocol for the MPML-DRM-A using Chinese remainder theorem. The proposed scheme ensures that 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, since the size of the watermark signal embedded

TABLE 3
Watermark Detection

Image	2 Watermarks $\alpha = 4$, $W_{len} = 109$			10 Watermarks $\alpha = 2$, $W_{len} = 81$		
	p	$n_{0.60}$	$n_{0.75}$	p	$n_{0.60}$	$n_{0.75}$
Watermark Strength ≈ 11.9467 and PSNR ≈ 36.1841						
Lena	1	2	2	0.8715	10	10
Baboon	1	2	2	0.0302	8	3
Pepper	1	2	2	0.0687	10	5
Boat	1	2	2	0.0027	3	2
Clown	1	2	2	0.4457	10	10
Barbara	1	2	2	0.0936	10	7
Goldhill	1	2	2	0.1277	10	7
Airplane	0.9103	2	2	0.0017	3	2
Fruit	1	2	2	0.0710	9	7
Watermark Strength ≈ 7.2207 and PSNR ≈ 40.9101						
Image	2 Watermarks $\alpha = 3$, $W_{len} = 65$			10 Watermarks $\alpha = 1$, $W_{len} = 109$		
	p	$n_{0.60}$	$n_{0.75}$	p	$n_{0.60}$	$n_{0.75}$
Lena	1	2	2	0.0009	3	2
Baboon	0.8692	2	2	0.0006	3	2
Pepper	0.8808	2	2	0.0012	3	2
Boat	0.7413	2	2	0.0004	3	2
Clown	1	2	2	0.0017	4	3
Barbara	0.9694	2	2	0.0018	3	2
Goldhill	1	2	2	0.0040	4	2
Airplane	0.5917	2	2	0.0003	3	2
Fruit	0.9244	2	2	0.0005	3	2
Watermark Strength ≈ 4.9668 and PSNR ≈ 43.1640						
Image	2 Watermarks $\alpha = 2$, $W_{len} = 85$			10 Watermarks $\alpha = 1$, $W_{len} = 65$		
	p	$n_{0.60}$	$n_{0.75}$	p	$n_{0.60}$	$n_{0.75}$
Lena	0.9650	2	2	0.0298	7	6
Baboon	0.4817	2	0	0.0005	3	2
Pepper	0.5579	2	1	0.0012	3	3
Boat	0.2801	0	0	0.0004	3	2
Clown	0.7890	2	2	0.0025	4	2
Barbara	0.6781	2	2	0.0013	3	2
Goldhill	0.6830	2	2	0.0007	3	2
Airplane	0.2801	0	0	0.0004	3	2
Fruit	0.6027	2	2	0.0006	3	2

into the content is independent of the number of distributors involved, the quality of the content used by all the consumers will be the same. The experiments performed in Section 5, clearly show the advantage of the joint watermark mechanism compared to the naive approach. The protocol takes care of the security concerns of owner, distributors and consumers. The identity of all the participants are carefully embedded into the content using the Chinese remainder theorem. The identity of all the participants can be determined from the watermark signal by reverse computing the CRT equations. In case the owner or distributors find an unauthorized copy, they can identify the traitors with the help of a judge.

As a future direction of research, the protocols may be improved to reduce the dependence on the license server. Further, in the proposed scheme the individual watermark information are computed as digital signatures. The protocols can be made more computationally efficient if these are replaced by any other easily verifiable watermark informations.

ACKNOWLEDGMENTS

Thanks to the Agency for Science, Technology and Research (A*STAR), Singapore for supporting this work under the project 'Digital Rights Violation Detection for Digital Asset Management'. Also, we thank the anonymous reviewers for their valuable comments.

REFERENCES

[1] R. J. Anderson, C. Manifavas, "Chameleon-A New Kind of Stream Cipher",Proc. 4th Int. Workshop Fast Software Encryption, London, U.K., 1997, pp. 107-113.
[2] D. Boneh, H. Shacham, B. Lynn, "Short Signatures From the Weil Pairing", Jornal of Cryptology, Vol. 17, No. 4, pp. 297-319, 2004.
[3] M. U. Celik, A. N. Lemma, S. Katzenbeisser, M. van der Veen, "Lookup-Table-Based Secure Client-Side Embedding for Spread-Spectrum Watermarks", IEEE TIFS, Vol. 3, No. 3, pp. 475-487, 2008.
[4] Guang-Huei Chiou, Wen-Tsuen Chen, "Secure Broadcasting Using the Secure Lock", IEEE Transaction on Software Engineering, Vol. 15, No.8, pp. 929-934, August 1989.
[5] I. J. Cox, J. Kilian, T. Leighton, T. Shamoon, "Secure Spread Spectrum Watermarking for Multimedia", IEEE TIP, Vol. 6, pp. 1673-1687, 1997.
[6] M. Deng, B. Prenel, "On Secure and Anonymous Buyer-Seller Watermarking Protocol", ICIW, pp. 524-529, 2008.
[7] S. Emmanuel, M. Kankanhalli, "Copyright Protection for MPEG-2 Compressed Broadcast Video", Proceedings. IEEE Int. Conference on Multimedia and Expo, 2001, pp. 206-209, 2001.
[8] Chun-I Fan, Ming-Te Chen, Wei-Zhe Sun, "Buyer-seller Watermarking Protocols with Off-line Trusted Third Parties", IJAHUC, Vol. 4, No.1 pp. 36 -43, 2009.
[9] F. Hartung, B. Girod, "Watermarking of Uncompressed and Compressed Video", Signal Processing, Vol. 66, No. 3, pp.283-301, 1998.
[10] F. Hartung, M. Kutter, "Multimedia Watermarking Techniques", Proceedings of the IEEE, Vol. 87, No. 7, pp.1079-1107, July 1999.
[11] 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.
[12] S. O. Hwang, K. S. Yoon, K. P. Jun, K. H. Lee, "Modeling and Implementation of Digital Rights", Journal of Systems and Software, Volume 73, Issue 3, pp 533-549, 2004.
[13] H. Jin, J. Lotspiech, "Attacks and forensic analysis for multimedia content protection", Int. Conference on Multimedia and Expo 2005.
[14] D. Johnson, A. Menezes, S. Vanstone, "The Elliptic Curve Digital Signature Algorithm (ECDSA)", http://www.comms.scitech.susx.ac.uk/fft/crypto/ecdsa.pdf.
[15] 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.
[16] W. Kanjanarin, T. Amornraksa, "Scrambling and Key Distribution Scheme for Digital Television", ICON, Proceedings of the 9th IEEE International Conference on Networks, pp.140-145, 2001.
[17] S. Katzenbeisser, A. Lemma, M. U. Celik, M. van der Veen, M. Maas,. "A Buyer-Seller watermarking protocol based on secure embedding", IEEE TIFS, Vol.3, No.4, pp. 783-786, 2008.
[18] D. Kundur, K. Karthik, "Video Fingerprinting and Encryption Principles for Digital Rights Management", Proc. IEEE, vol. 92, no. 6, pp. 918-932, Jun. 2004.
[19] N. Memon, P. W. Wong, "A Buyer Seller Watermarking Protocol", IEEE TIP, Vol. 10, No. 4, pp.643-649, 2001.
[20] A. Menezes, P. van Oorschot, S. Vanstone, *Handbook of Applied Cryptography*, CRC-Press, page 610-612, 1996.
[21] H. Muratani, "Optimization and Evaluation of Randomized c-Secure CRT Code Defined on Polynomial Ring", IH 2004, Canada, pp. 282-292, May 23-25, 2004.
[22] A. Sachan, S. Emmanuel, A. Das, M. Kankanhalli, "Privacy Preserving Multiparty Multilevel DRM Architecture", CCNC 2009.
[23] T. Thomas, S. Emmanuel, A. Das, M. S. Kankanhalli, "Secure Multimedia Content Delivery with Multiparty Multilevel DRM Architecture. NOSSDAV 2009.
[24] FIPS 180-2: Secure Hash Standard (SHS).
[25] Z. Wang, Bovik, A.C, "Mean squared error: Lot it or leave it? A new look at Signal Fidelity Measures", IEEE Signal Processing Magazine, Vol: 26 Issue: 1, pp. 98-117, 2009