國立交通大學

資訊科學與工程研究所

碩 士 論 文

一個具匿名性與可轉移授權的數位權利管理系統

An Anonymous and Authorization Transferable Digital Rights Management System

研 究 生：陳仕烽

指導教授：曾文貴　教授

中 華 民 國　九 十 六　年 六 月

一個具匿名性與可轉移授權的數位權利管理系統

An Anonymous and Authorization Transferable Digital Rights Management System

研 究 生：陳仕烽　　Student：Shih-Feng Chen

指導教授：曾文貴　　Advisor：Wen-Guey Tzeng

國 立 交 通 大 學

資 訊 科 學 與 工 程 研 究 所

碩 士 論 文

A Thesis

Submitted to Institute of Computer Science and Engineering

College of Computer Science

National Chiao Tung University

in partial Fulfillment of the Requirements

for the Degree of

Master

in

Computer Science

June 2007

Hsinchu, Taiwan, Republic of China

中華民國九十六年六月

一個具匿名性與可轉移授權的數位權利管理系統

學生: 陳仕烽　　指導教授: 曾文貴 教授

國立交通大學資訊科學與工程研究所碩士班

摘　　要

隨著網際網路與數位科技的發展，消費者可以利用各種傳輸方式取得各種數位內容。數位內容極易複製，快速傳輸與大量散佈，這些性質帶給內容提供者一個棘手的問題，要如何防止產品被不當的大量散佈給其他沒有經過適當授權的使用者? 可以利用數位權利管理系統(Digital Rights Management，DRM)，經由 DRM 系統控制並設定存取權限，使得只有經過適當授權的使用者，才能夠存取這些受保護的內容，避免數位內容被不當的散佈與存取，以保障著作人及出版商應享有的權利。

但從另外一個角度來看，現今大多的 DRM 系統在設計時，偏重於以系統商的利益為出發點，往往過於縮減限制使用者應有的權利，或是私下收集使用者資料而侵犯隱私。我們以使用者要求"合理使用"的理由為出發點，提出一個具有匿名性及可轉移授權機制的 DRM 系統。使用者在正常使用下享有匿名性，且能自由傳輸授權給予其他使用者；而對系統而言，非法散佈授權的使用者則會喪失匿名性，系統能偵測出其身分，以達到在系統商的利益與消費者之間權利取得一個較佳的平衡點。

關鍵字: 數位權利管理, 合理使用.

An Anonymous and Authorization Transferable Digital Rights Management System

Student: Shih-Feng Chen Advisor：Dr. Wen-Guey Tzeng

Institute of Computer Science and Engineering

National Chiao Tung University

Abstract

With the Internet and digital multimedia technology development, the consumers could easily obtain the digital contents in variant ways. The content providers need some mechanisms to enforce access policies and to control how users to use their digital contents. Digital Rights Management (DRM) system is a system to protect and manage digital contents and control the usage and distribution of those digital contents. Only proper authorized user could access these protected digital contents.

However, most current DRM systems add too many restrictions on the right of users and may gather some unnecessary privacy information without the user's permission. We discuss the issue of "fair use" and purpose a DRM system with anonymity and authorization transferability. The honest user could anonymously access the system service and transfer his authorization to other users, and the system could trace the identity of the dishonest user with illegal distributing. We hope our system to achieve a balance between the rights of the owners and the accessibility of the consumers.

Keywords: Digital Rights Management (DRM), fair use.

致　謝

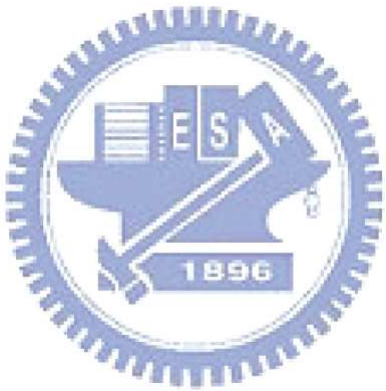
在此感謝我的指導老師曾文貴教授,在我碩士班的學習過程中,不只在學業上帶領我走進密碼學的領域，更在生活和言行舉止上孜孜不倦的教導我，使我受益良多。另外，我要感謝口試委員，交大資工蔡錫鈞教授，清大資工孫宏民教授及中央研究院資訊科學研究所呂及人教授，在論文上給我許多良好的建議與指導，讓我的論文更加完善。除此之外，我要感謝實驗室朱成康學長與林孝盈學姐的指導，以及實驗室同學劉易儒，周昆逸和學弟妹們在這段期間付出的一切協助。

最後，我要感謝我的家人,不論在精神或物質上都給我最大的支持，讓我在無後顧之憂的情況下可以順利完成學業。在此，僅以此文獻給所有我想要感謝的人。

Table of Contents

List of Figures

List of Tables



Chapter 1

Introduction

With the Internet and digital multimedia technology development, the consumers could easily obtain different kinds of digital contents (e.g. documents, books, music, and movies) in variant ways. As the analog media loses quality with each copy generation, oppositely, the digital contents could be copied or distributed to unlimited numbers without losing their quality. Therefore, the content providers need some mechanisms to enforce access policies and to control how users to use their digital contents.

Digital Rights Management (DRM) system is a system to protect and manage digital contents and control the usage and distribution of those digital contents. The DRM technology is to protect and manage digital contents and control the usage and distribution of those digital contents. With DRM technologies, content providers and copyright holders could protect their works from illegal use, and only proper authorized users could access these protected digital contents.

However, the design concept is based on the right holders' benefit in most current DRM systems. It adds too many restrictions on the right of users and may gather some unnecessary privacy information without the user's permission. We discuss the issue of "fair use" and purpose a DRM system with anonymity and authorization transferability. The honest user could anonymously access the system service and transfer his purchased content and the corresponding rights to other users. If a dishonest user has illegally transferred his purchased contents and rights to multi users, the system could trace the identity of the dishonest user with illegal distributing. We hope to achieve a balance between the rights of the owners and the accessibility of the consumers.

The rest of the thesis is organized as follows. In chapter 2, we describe the content of DRM systems and related cryptography theories and techniques. In chapter 3, we propose a DRM system with properties of anonymity and transferability and discuss the implementation

of our system. In chapter 4, we analyze the security of our scheme. Finally, we conclude the thesis and indicate the future work in chapter 5.

Chapter 2

Preliminaries

In this chapter, we first introduce the background and structure of DRM system, introduce current DRM systems in online music market, and we will discuss the concept about ”fair use” in section 2.1. Then we introduce zero-knowledge proof system of knowledge in section 2.2 and group signature scheme in section 2.3.

2.1 Digital Rights Management

Digital Rights Management (DRM) system is a system to protect and manage digital contents and control the usage and distribution of those digital contents. With DRM technologies, content providers and copyright holders could protect their works from illegal use, and only proper authorized user could access these protected digital contents.

2.1.1 Background

As the analog media loses quality with each copy generation, the different kinds of digital contents (e.g. documents, books, music, and movies) could be copied or distributed to unlimited numbers without losing their quality. The content providers need some mechanisms to enforce access policies and to control how users to use the digital contents. The DRM technology is to protect and manage digital contents and control the usage and distribution of those digital contents. It is not only focused on security to unauthorized copy, which is to protect the digital contents with encryption, and could not prevent some authorized users from distributing their decrypted data to other unauthorized users. The DRM system should protect, monitor, and track all forms of rights usages [1] [12] [15].

DRM is widely used by the entertainment industry, such as online music stores (like Apple iTunes Music Store [24] and Napster music store [33]), some pay-TV producers, and also to protect documents in enterprises [23]. Different DRM vendors have different DRM

implementations. In general, there are three important players involved in a DRM system: the consumer, the producer and the publisher [15]. We will give the expansions in detail below. The roles in a simplified DRM system are illustrated in Figure 1.

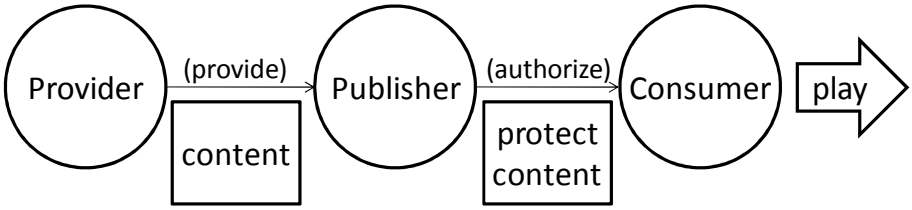


Figure 1 The roles in a DRM system

- The content **Provider** produces content and holds the digital rights of the content. The provider (an author or a record label) wants to protect these rights and sell content to the publisher.
  - Content means different kinds of data to protect. It can be document, book, audio, video, etc..
- The **Publisher** manages the DRM system used to control and distribute digital content. The DRM system encrypts the content, packages digital rights in a license to the corresponding content, and distributes the protected content to the consumers.
  - The License is digital data that specifies certain usage rules for the digital content. There may be different licenses corresponding to the same content. The usage rules could be the expiration time, copy permission, or the access frequency of the content. The license can be expressed by the form of the Right Object.
- The **Consumer** obtains the protected digital content from the publisher and then paying for the digital license to get the rights to access protected digital content.
  - The Right refers to the usage rules in a license for the digital content, which means:

    Principal -under- *Condition* -apply- *Rights* -to- *Resource*

    The right is usually defined and represented in the rights expression language (REL). The most common REL is *Extended Rights Markup Language* (XrML)

[1] [29]. It is a XML-based language for describing rights.

☐ Cryptographic Mechanisms in a DRM system

Several cryptographic primitives are commonly used in DRM systems. The content provider may send the digital content with digital signature technology to ensure the integrity of content. The publisher could encrypt a digital content by using standard cryptographic encryption mechanisms. Only an authorized purchaser who can authenticate himself as trustworthy could get the corresponding digital license. Authorized user could verify the content-license pair by signature and hash function, and then decrypt the digital content with the key in his license.

The providers and publishers need more protection after content decryption. If the authorized user saves the content in an unprotected form and distribute the digital copy on the Internet, other unauthorized users could access content without any restriction. Therefore, we need some detection mechanism of protected content to trace the usage, which can be obtained by a combination of digital signature, certificate, content fingerprinting and embedding watermarks. Before the content is encrypted, identifying data signed by the consumer can be embedded as watermark in the content.

2.1.2 **Current DRM Systems**

In this subsection, we focus on the main DRM systems used in online music market.

☐ Microsoft WMRM

Microsoft Windows Media Rights Management (WMRM) is a DRM system to protect and securely deliver a subscription content for playback on a computer, portable device, or network device [35]. The main advantage of WMRM is that the Windows media format is widely used on the Internet, and the Windows media player has already incorporated DRM support [14]. The newest version is WMRM 10.

There are 3 main roles in WMRM system: content provider, license issuer, and user.

Figure 2 shows how content is protected, distributed, and used with WMRM:

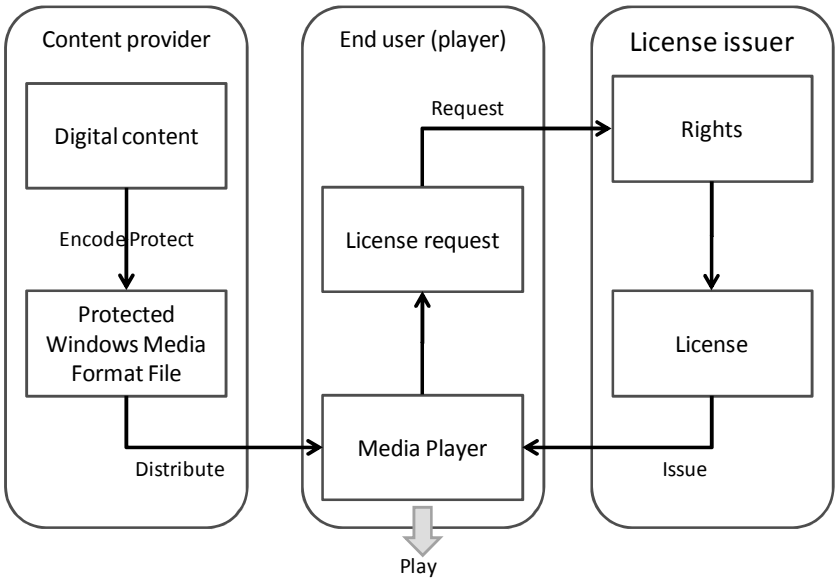


Figure 2 Windows Media Right Manager DRM process

1. The content provider encodes and packages digital contents into Windows Media Format files. During the encoding process, the DRM profile adds certain information to the content header, and the content is encrypted.
2. The content provider delivers packaged Windows Media files to user. The license can be delivered with file, or it can be delivered after a user provides additional information.
3. When the user tries to play a protected file, the player searches the user's computer for a valid license.
4. If the user's player fails to find the necessary license, it sends a license request which contains the header of the protected content and the user's system information to the license issuer.
5. The license issuer issues licenses for Windows Media files. The licenses contain the encrypted content key, and other properties that specify the use of the Windows Media file.

The licenses describe the rights that specify the use of the Windows Media file. The content provider sets these rights to determine which actions are allowed from minimal

control over playback to more restrictive licenses. We list some important rights supported in WMRM in Table 1:

Right	Name
Playback, Copy, Transfer	Playcount
	AllowCopy
	CopyCount
	TransferCount
Burning to CD	AllowPlaylistBurn
	MaxPlaylistBurnCount
	PlaylistBurnTrackCount
Playback Time	BeginDate
	ExpirationDate
	ExpirationAfterFirstUse
	ExpirationOnStore

Table 1 Important rights supported in WMRM.

Napster music store [33] is a subscription based online music service with WMRM DRM technology. The user could download an unlimited amount of music while subscribed to the service. The user could use the music on his portable device for an additional $5 per month. Once the user stops subscription to the service, all music downloaded will be unusable. If user wants to burn a track to CD, he has to pay an additional $.99.

☐ Apple FairPlay

FairPlay is a DRM technology created by Apple Inc., and used by Apple's products, for online music service: iTunes Store, for multimedia player software: iTunes Jukebox, and for multimedia portable device: iPod [24]. All media downloaded from iTunes Store are protected with FairPlay.

The FairPlay-protected audio files are encoded in 128 kbit/s AAC (Advanced Audio Coding) format. The protected files can only be played on iTunes Jukebox, iPod, and a few

compatible portable devices. However, the protected files are not supported by other software jukebox like Microsoft Media Player or most portable players. The protected audio files have the restrictions including:

- The protected files can be accessed on up of five authorized computers simultaneously.
  - If the user wants to play his purchased songs on the sixth computer, he needs to deauthorize one of the five authorized computers before.
  - The user could deauthorize all the computers in his account once per year.
- The users can burn the protected files to a standard Audio CD for any number of times.
  - The user can copy a particular playlist containing files protected with FairPlay to a CD up to seven times.
- The users can transfer the protected files to any number of iPod portable devices.
  - However, the protected files can not be transferred from the iPod to another unauthorized computer.

On April, 2007, Apple's iTunes Store had sold more than 2.5 million songs, and it is the world's most popular online music, TV and movie store [25]. On May, 2007, Apple launched iTunes Plus, a DRM-free online music service [26]. The iTunes Plus tracks have higher quality without any usage restrictions. The price of original iTunes track is $0.99, and the Plus version is $1.29. The file format is upgraded to 256 kbits/s AAC streams for better sound quality. Until now, only music tracks from EMI are provided in iTunes Plus DRM-free version.

2.1.3 **Fair Use**

From the provider's and publisher's viewpoint, the DRM system could ideally protect their contents from illegal use. All consumers are only allowed to do all the authorized activities in licenses. In other words, the consumers could not make unauthorized or

undefined use of their purchased contents. It may hinder interoperability of public knowledge, likes the library collections and the academic research. The library may get into trouble to make digital copies for some digital content with uncertain rights. However, REL or any other ways for rights definition is impossible to cope with all situations.

For instance, a consumer has bought a track from online music store. It is without reason to put any restrictions on playing at his place, such as his office or his home. Another example is that it is impossible for a consumer to resell or send the track to his friend in most systems. In practical, it is hard to distinguish a legal backup from illegal copy very clearly.

☐ DRM and Copyright Law

The Digital Millennium Copyright Act (DMCA) [28] is the legislation that implements two 1996 World Intellectual Property Organization (WIPO) treaties: the WIPO Copyright Treaty and the WIPO Performances and Phonograms Treaty. According to DMCA, any attempt for the creation and distribution of DRM circumvention tools even for legal reasons may violate federal law under DMCA [14].

In Taiwan, fair use is backed by Article 65, 80-1 and 80-2, paragraph 3, of Copyright Act [27], which gives the following facts shall be noted as the basis for determination:

1. The purposes and nature of the exploitation, including whether such exploitation is of a commercial nature or is for nonprofit educational purposes.
2. The nature of the work.
3. The amount and substantiality of the portion exploited in relation to the work as a whole.
4. Effect of the exploitation on the work's current and potential market value.

We notice that the utilization of "fair use" is case by case with flexible rules. Copyright Act also gives some exceptions to copyright which allow users to reproduce and excerpt copyrighted works for purposes including: to protect personal data, for file archive institutions, educational institutions, or public libraries to assess whether to obtain the information.

Fair use is a privilege which could not be deprived by the license agreement. However, it

is still difficult to address the reasonable amount and substantiality in law to fit all situations. When the content providers and copyright holders apply DRM technologies to their applications, they should notice fair use issue to prevent their consumers from violating the license agreement.

☐ Transfer of Ownership

Another issue is to transfer the ownership [1][11][16]. The ownership transferability is defined in Article 59-1of Copyright Act [27], without the copyright holders' permission or authorization. However, it is limited to computer program works but multimedia contents like music and movies. From content provider's perspective, the consumer's ability to share a copy of content with other people should be fully restricted. When the consumer downloads digital content from the publisher, he only obtains authorized usage rights by license agreements. The consumers do not obtain the ownership of contents from license agreements, so they are not authorized to transfer the usage rights to the third parties.

There are many online music service providers including Apple's iTunes Store [24] (except iTunes Plus), Yahoo! Music [36], Napster [33], Wal-Mart Music Downloads [34], ezPeer+ [30], KKBOX [32], and all of them do not provide transferability. It dose not only means to download digital contents to user's portable devices but also to transfer the content and the authorization to the third parties. We consider that it is reasonable for the DRM systems should implement transferability.

Obviously, if the DRM system gives fewer restrictions to the consumers, it could attract more consumers. Apple's FairPlay DRM is a successful example. The purchasers could download the content to unlimited numbers of portable device, and they could play the content on five computers registered to the same account. It is easier to share the content to their family according to the right of non-public. It is allowed of a small mount share for personal use, but massive share will infringement content provider's benefit. As a result, the concept of "to transfer" seems to meet the content provider's need more than "to share". The total number of copies is restricted, and the consumer will lose their rights after they transfer

their authorization to other consumers.

In summary, we argue that the DRM system should make some necessary concessions to provide transferability.

☐ Privacy in DRM

One of the major issues raised by DRM system concerns the protection of the user's privacy and anonymous consumption of digital content [16]. When the consumer accesses the protected content, the consumer needs to reveal his identity and send his personal information to the server in process of authentication. Some DRM systems may collect more than necessary privacy information in secret and the users never know about how the usage of these information. They may be send collected user-specific information to marketing agencies without the user's permission. It will cause some privacy problem.

For example, Apple's iTunes Store collect and store the following personal data during concluding a purchase [12]:

- The client OS, iTunes software version, and client IP address
- The iTunes client software id called Device-id
- The user's e-mail address called Apple-id
- The Product-id and meta data

The iTunes server will utilize the information to encrypt the content. Therefore, only the registered user with registered device could access the protected data. But it is doubted whether sending these data are all necessary.

While Apple starts iTunes Plus service, it provides a DRM-free environment without any restrictions. However, the iTunes Plus track still includes the original purchaser's account inside the ACC header. It just likes a light-weight DRM system. Everything in the scope of fair use can be done in the way users are familiar with. When the DRM-free track is distributed to the public, for example a P2P file share system, the owner of the track can be traced.

A good DRM system should grant anonymous access to the digital content. It is

recommended to do all personal data processing in a clear transparent mode [12]. The consumers have the rights to know how their privacy information utilized. If they do not accept, they can stop the service.

In conclusion, we address the issue of transfer of rights and user anonymity in this thesis.

2.2 Zero-Knowledge Proof System

To archive anonymity, we make use of group signature, which will be introduced in next subsection, and “proof systems" that allow one party to convince other parties about its knowledge of certain values like the membership, such that no useful information is leaked.

We introduce various zero-knowledge protocols for proving knowledge of discrete logarithms:

- $\{\mathrm{PK}\{(\alpha) : y = g^{\alpha}\}$

  Proving the knowledge of a discrete logarithm x of a group element y to a base g [19].

  1. The prover chooses a random $r \in_R \mathbb{Z}_Q$, computes $t := g^r$ and sends t to the verifier.
  2. The verifier picks a random challenge r and sends r to the prover.
  3. The prover computes $s := r - cx \pmod{Q}$ and sends s to the verifier.
  4. The verifier accepts, if $g^s y^c = t$ holds.

  We can convert the interactive proof into a non-interactive proof. Let $H : \{0,1\}^* \to \mathbb{Z}_Q$ be a collision-resistant hash function.

  1. The prover computes the challenge $c := H(g|y|g^r)$, sets $s := r - cx$, and sends (c, s) to the verifier.
  2. The verifier accepts if $c = H(g|y|g^s y^c)$ holds.

- $\mathrm{PK}\{(\alpha) : y_1 = g^{\alpha} \wedge y_2 = h^{\alpha}\}$

  Proving the equality of the discrete logarithms of elements y_1 and y_2 to the bases g and h, respectively (proof of AND) [8].

1. Let $x = \log_g y_1 = \log_h y_2$.
2. The prover chooses a random $r \in_R \mathbb{Z}_Q^*$, computes $t_1 := g^r, t_2 := h^r$, and sends t_1, t_2 to the verifier.
3. The verifier picks a random challenge c and sends c to the prover.
4. The prover computes $s := r - cx \pmod Q$ and sends s to the verifier.
5. The verifier accepts, if $g^s y_1^c = t_1 \wedge h^s y_2^c = t_2$ holds.

To convert the interactive proof into a non-interactive proof:

1. The prover computes the challenge $c := H(g|y_1|h|y_2|g^r|h^r)$, sets $s := r - cx$, and sends (c, s) to the verifier.
2. The verifier accepts if $c = H(g|y_1|h|y_2|g^s y_1^c|h^s y_2^c)$ holds.

☐ $PK\{(\alpha, \beta) : y_1 = g^\alpha \vee y_2 = h^\beta\}$

Proving the knowledge of (at least) one out of the discrete logarithms of the elements y_1 and y_2 to the base g (proof of OR) [9] [18].

W.l.o.g., we assume that the prover knows $x = \log_g y_1$.

1. The prover chooses random $r_1, s_2 \in_R \mathbb{Z}_Q^*, c_2 \in_R \mathbb{Z}_Q$, computes $t_1 := g^{r_1}, t_2 := g^{s_2} y_2^{c_2}$, and sends t_1, t_2 to the verifier
2. The verifier picks a random challenge c and sends it to the prover.
3. The prover computes $c_1 := c \oplus c_2$ and $s_1 := r_1 - c_1 x \pmod Q$(where $\oplus$ denotes the bit-wise XOR operation) and sends (s_1, s_2, c_1, c_2) to the verifier.
4. The verifier accepts, if $c = c_1 \oplus c_2$ and $t_i = g^{s_i} y_i^{c_i}$ for $i \in \{1,2\}$ holds.

The interactive proof can be converted into a non-interactive proof:

1. The prover randomly chooses $r_1, s_2 \in_R \mathbb{Z}_Q^*, c_2 \in_R \mathbb{Z}_q$, computes the challenge $c := H\left(g|y_1|h|y_2|g^{r_1}\middle|g^{s_2} y_2^{c_2}\right)$, computes $c_1 := c \oplus c_2$, sets $s_1 := r_1 - c_1 x_1, s_2 := r_2$ and sends (s_1, s_2, c_1, c_2) to the verifier.
2. The verifier accepts, if $c = H\left(g|y_1|h|y_2|g^{s_1} y_1^{c_1}\middle|h^{s_2} y_2^{c_2}\right)$ holds.

We can extend the above approaches for proving arbitrary monotone statements built with $\wedge$'s and $\vee$'s.

2.3 Group Signature

Group Signature was first introduced in [6] and provides anonymity for signers. Only the member of the group can sign messages, the verifier can only tell someone of the group members signed, and the signer's identity is kept secret. In some group signature schemes, there is a party with the ability to trace the signature or revoke the anonymity [2][4][5]. The group signature scheme could be applied to the DRM system. All users are registered to the DRM system which as a group manager, and when they want to access the protected content, they can authenticate their identity with applying a group signature.

A group signature scheme must satisfy the following properties: correctness, unforgeability and anonymity. The correctness property means that the signatures generated by honest signer can be verified and trace correctly. The unforgeability means that only group members can generate valid signature. The anonymity means that no one is able to find out which group member signed a message or to decide whether two signatures have been issued by the same group member.

2.3.1 *k*-Times Anonymous Authentication Scheme

The *k*-Times Anonymous Authentication (*k*-TAA) scheme is proposed in 2004 [21]. It is a modification of a group signature scheme and allows the group members to be authenticated anonymously by an allowable number of times. If a group member exceeds the time restriction, his identity will be public traced from an authentication log by any verifier. In a *k*-TAA system, there are three participants, a group manager (GM), an application provider (AP), and the users. The group manager first registers all users, then the application provider publish the number of times allowed a user to authenticate, and the registered users can be authenticated by the AP.

☐ Key Ideas

Let the AP set the bound at *k*. A registered user has *k* different basis $B_1, \ldots, B_k$ for the AP, which is called a tag base. When he wants to be authenticated by the AP, he picks a fresh

tag base B_i not used before. If he uses different tag bases, his identity will not be traced. However, if he has authenticated for k+1 times, he must have showed the same tag base twice, and anyone can trace his identify.

☐ k-TAA Scheme

There are five phases in the k-TAA scheme:

1. **Setup**

   GM randomly chooses a rigid integer n, $(a, a_0) \in QR(n)^2$, $g \in G$.

   The group public key is (n, a, a_0, g).

2. **Joining**

   User U_i chooses a random x, computes a secret/public key pair $(sk, pk) = (x, (A, e, \alpha, ID))$, such that $a^x a_0 = A^e, (\alpha, ID) = (a^x, g^x)$.

   GM adds (i, ID) to the identification **LIST**.

3. **Bound announcement**

   AP publishes k different tag bases: $(t_i, \check{t}_i) \in G^2$ for $i = 1, \dots, k$

4. **Authentication**

   AP sends a random number c to U_i.

   U_i selects a fresh tag base $(t_i, \check{t}_i)$ and computes tag $(T, \check{T}) = (t_i^x, (g^c \check{t}_i)^x)$.

   U_i sends back the tag $(T, \check{T})$ to AP.

   AP adds $(c, (T, \check{T}))$ to the authentication log **LOG**.

5. **Public tracing**

   If user has repeated the same tag base, there must be two records $(r, (T, \check{T})), (r', (T', \check{T}'))$ with the same tag $T = T'$ in **LOG**.

   Anyone can trace the user's identity by computing $ID = \left(\frac{\check{T}'}{\check{T}}\right)^{\frac{1}{c'-c}} = \left(\frac{(g^{c'}\check{t}_i)^x}{(g^c \check{t}_i)^x}\right)^{\frac{1}{c'-c}} = b^x$ from **LOG** and **LIST**.

In our purposed scheme, we set the bound $k = 1$. The registered user hold only one tag

base for every license to make sure there is only one user access the service simultaneously. We can also apply *k* different tag bases to a specific RO for *k* users, but it will seriously add the verifier's overheads, and the mounts of valid content may grow exponentially. Therefore, we would rather set each *k* ROs with a single tag base than a single RO with *k* tag bases.

Chapter 3

An Anonymous and Authorization Transferable DRM System

In this chapter, we first introduce the background and structure of DRM system in section 3.1. Then we explain the main idea about the system construction in section 3.2. We give our purposed scheme in section 3.3 and discuss our DRM system implementation in section 3.4.

3.1 Introduction

In the previous chapter, we have talked about the concept of "fair use". We consider two requirements in current popular DRM systems, transferability and anonymity. Transferability means that a registered user could transfer not only the content but also the corresponding license to the other users. Take Apple's iTunes for example, if the transferability is implemented, the users can directly transfer their authorization to other users without online deauthorization. For achieving anonymity, the DRM system is only allowed to collect the necessary information to protect personal privacy.

However, beyond the anonymity, the users may illegally distribute content and license. Therefore, we won't protect dishonest users, and we will disclose the anonymity for the purpose of restricting illegally transmitting authorization. If a dishonest user after transferring his authorization has continued to access the service, the user will be traced his identity by system and lose his anonymity to protect the author and right holder's benefit.

A trivial way to solve the transferability is to embed the sender's identity information or digital signature into the protected contents. But it will conflict with the user's anonymity. While we hope to hold the anonymity in the normal DRM system, we hope that the DRM system with transferability can not trace the users' identity in the transfer chain if all users are honest.

3.2 System Construction

We adopt a modified version of *k*-TAA scheme [21] and a transferable E-Cash model purposed in [7] and [13] to construct our purposed system.

The participants in the scheme are the register server S, the content server AP, and the user U. As the original *k*-TAA scheme, the register server S registers all users, and the registered users can be authenticated by the content server AP.

The **Register** protocol is as same as the original **Joining**, and original **Bound announcement** is modified to **Sign** and **Issue**, to force users to use valid tag bases to the corresponding Right Object and to support various payment systems. We extend the original **Authentication** in *k*-TAA scheme to two protocols, **Transfer** and **Authentication**. While user U_i wants to transfer his authorization to other user U_j, U_j plays the role as AP in the original *k*-TAA scheme. The receiver U_j should keep the authentication chain history to next user or AP. If a dishonest user U_k has transferred to multi parties, there will be two records with the same prefix after AP receiving both chains. Finally, AP could apply the **TRACE** algorithm to identity who has transferred to multi parties.

☐ Register

As the original **Joining** protocol, the user U_i randomly chooses his secret key u_i and then registers his identity ID_i to the register server S. S adds ID_i to a public identification **LIST**.

☐ Sign

U_i anonymously proves that he is one of the registers in **LIST** to S. If S accepts the proof, U_i gets a tag token t_i from S.

☐ Issue

U_i sends the tag token t_i to AP and gets Right Object **RO** associated to the tag token t_i. U_i initializes transfer chain $R_{si} = (\mathbf{RO}, \mathrm{hash}(t_i))$. Since we tie the hash value of the tag token t_i into R_{si}, U_i has to use the same t_i and R_{si} in **Transfer** or **Authentication**

protocol.

☐ Transfer

U_i sends R_{si} and corresponding t_i to U_j and proves knowledge of t_i. U_j chooses a tag token t_j and computes a challenge $c_{ij} = \text{hash}(t_j)$ and sends c_{ij} to U_i. Next, U_i computes response r_{ij}, and sends r_{ij} to U_i. Finally, U_j gets extended transfer chain $R_{sj} = R_{si}||(t_i, c_{ij}, r_{ij})$. The transfer chain grows in size as it changes more hands. All users should show the same tag token which used to exchange the transfer chain to the next party. The transfer chain takes on the form: $R_{sz} = (\mathbf{RO}, \text{hash}(t_i))||(t_i, c_{ij}, r_{ij})|| \dots ||(t_y, c_{yz}, r_{yz})$.

☐ Authentication

U_j sends R_{sj} and the same token t_j used for R_{sj} to AP and proves knowledge of t_j. AP verifies the proof and the transfer chain R_{sj} and accepts if it passes all verification. AP adds the transfer chain R_{sj} and the proof to the authentication **LOG**.

☐ Trace

If the dishonest user U_i has transferred to two user U_j and U_k, after both of U_j and U_k authenticated to AP, there must be two records R_{sj} and R_{sk} with the same prefix $(\mathbf{RO}, \text{hash}(t_i))|| \dots ||(t_i, c_{ij}, r_{ij})$ and $(\mathbf{RO}, \text{hash}(t_i))|| \dots ||(t_i, c_{ik}, r_{ik})$ in the authentication **LOG**. Since we construct t_i and r_{ij} as a tag base and tag pair in *k*-TAA scheme. Therefore, AP can directly apply the Trace algorithm on (t_i, c_{ij}, r_{ij}) and (t_i, c_{ik}, r_{ik}) to find out the dishonest user's identity ID_i from **LIST**.

3.3 Our Purposed Scheme

Let $\mathcal{H}: \{0,1\}^* \to \mathbb{Z}$ be a collision-resistant hash function, $G = \langle g \rangle = \langle h \rangle$ be a cyclic group of order q on which DDH problems are hard to solve. g and h are elements of G. $\Sigma = (\text{KeyGen}, \text{Sign}, \text{Verify})$ is an unforgeable signature scheme

3.3.1 **Setup**

1. S: generate sign/verify key pair $\mathrm{KeyGen}(1^k) \rightarrow (sk_s, vk_s)$.
2. AP: generate sign/verify key pair $\mathrm{KeyGen}(1^k) \rightarrow (sk_{AP}, vk_{AP})$.

3.3.2 **Register**

The user U_i registers ID_i and proves the knowledge of secret key to S.

1. U_i: choose a random $u_i \in_R \mathbb{Z}_q^*$.
2. $U_i \rightarrow S$: $(i, ID_i) = (i, g^{u_i})$.
3. $U_i \leftrightarrow S$: $PK\{(\alpha): ID_i = g^{\alpha}\}$.
   i. U_i: choose a random $r \in_R \mathbb{Z}_q$.
   ii. U_i: compute the challenge $c \coloneqq \mathcal{H}(g||ID_i||g^r)$.
   iii. U_i: set $s \coloneqq r - cu_i$, and send (c, s) to S.
   iv. S: accept the proof if $c = \mathcal{H}(g||ID_i||g^s ID_i^c)$ holds. Otherwise, send an error message to U_i and abort.
4. S: add (i, ID_i) to **LIST**.

3.3.3 **Sign**

The user U_i anonymously proves he is one of the registers, and get a tag token t_i.

1. U_i: choose a random number $w_i \in_R \mathbb{Z}_q^*$ to mask his identity ID_i.
2. $U_i \rightarrow S$: $m_i = (m_{i,1}||m_{i,2}) = (ID_i^{w_i}||g^{w_i})$.
3. $U_i \leftrightarrow S$: $PK\left\{(u_i, w_i, ID_1, \dots, ID_n): \begin{matrix} (ID_1 = g^{u_i} \wedge m_{i,1} = ID_1^{w_i} \wedge m_{i,2} = g^{w_i}) \vee \dots \\ \vee (ID_n = g^{u_i} \wedge m_{n,1} = ID_1^{w_i} \wedge m_{n,2} = g^{w_i}) \end{matrix}\right\}$.
   i. Assume U_i know (u_i, w_i) such that $(ID_i = g^{u_i} \wedge m_{i,1} = ID_1^{w_i} \wedge m_{i,2} = g^{w_i})$ holds.
   ii. U_i: randomly pick $(r_{j,1}, r_{j,2}) \in Z_q^* \times Z_q^*$ for $j = 1, \dots, n$, and $c_j \in Z_q^*$ for $j \neq i$.
   iii. U_i: compute $t_{i,1} \coloneqq g^{r_{i,1}}, t_{i,2} \coloneqq ID_i^{r_{i,2}}, t_{i,3} \coloneqq g^{r_{i,2}}$, and $t_{j,1} \coloneqq g^{r_{j,1}} ID_j^{c_j}, t_{j,2} \coloneqq ID_j^{r_{j,2}} m_{i,1}^{c_j}, t_{j,3} \coloneqq g^{r_{j,2}} m_{i,2}^{c_j}$ for $j \neq i$.

iv. U_i: compute the challenge

$$c = \mathcal{H}(g||m_{i,1}||m_{i,2}||ID_1||\ ...\ ||ID_n||t_{i,1}||t_{i,2}||t_{i,3}||\ ...\ ||t_{n,1}||t_{n,2}||t_{n,3})$$

v. U_i: compute $c_i := c \oplus c_1 \oplus ... \oplus c_j \oplus ... \oplus c_n$, for $j \neq i$.

vi. U_i: set $s_{i,1} := r_{i,1} - c_i u_i, s_{i,2} := r_{i,2} - c_i w_i$, and $s_{j,1} := r_{j,1}, s_{j,2} := r_{j,2}$ for $j \neq i$.

vii. $U_i \rightarrow S$: $(s_{1,1}, s_{1,2}, \dots, s_{n,1}, s_{n,2}, c_1, \dots, c_n)$.

viii. S: accept the proof if

$c = \mathcal{H}(g||m_{i,1}||m_{i,2}||ID_1||\ ...\ ||ID_n||g^{s_{1,1}}ID_1^{c_1}||ID^{s_{1,2}}m_{1,1}^{c_1}||g^{s_{1,2}}m_{1,2}^{c_1}||\ ...)$ holds.

Otherwise, send an error message to U_i and abort.

4. $S \rightarrow U_i$: $\sigma_i = Sign_{sk_S}(m_i)$.

5. U_i: set tag token $t_i = (m_i, \sigma_i)$

3.3.4 **Issue**

The user U_i gets Right Object **RO** and initializes transfer chain R_{si}.

1. $U_i \rightarrow AP$: t_i

2. $U_i \leftrightarrow AP$: $PK\{(u_i, w_i): (m_{i,1} = m_{i,2}^{u_i} \wedge m_{i,2} = g^{w_i})\}$.

i. U_i: randomly pick $(r_1, r_2) \in_R \mathbb{Z}_q^2$.

ii. U_i: compute the challenge $c := \mathcal{H}(g||m_{i,1}||m_{i,2}||m_{i,2}^{r_1}||g^{r_2})$.

iii. U_i: set $s_1 := r_1 - cu_i, s_2 := r_2 - cw_i$, and send (c, s_1, s_2) to AP.

iv. AP: verify tag token t_i by $Verify(pk_S, m_i, \sigma_i)$ and accept the proof if $c = \mathcal{H}(g||m_{i,1}||m_{i,2}||m_{i,2}^{s_1}m_{i,1}^{c}||g^{s_2}m_{i,2}^{c})$ holds. Otherwise, send an error message to U_i and abort.

3. AP: according the content and license, properly set Right Object **RO**, and then generate a signature $\sigma_0 = Sign_{sk_{AP}}(\mathbf{RO}||hash(t_i))$. Send $(\mathbf{RO}, \sigma_0)$ to U_i.

4. U_i: initial transfer chain $R_{si} = (\mathbf{RO}||c_0), Proof_{si} = (\mathbf{RO}||hash(t_i)), \sigma_0, Proof_{si}$.

3.3.5 **Transfer**

The user U_i authenticate to U_j. After transfer, U_j gets extended transfer chain $R_{sj} = R_{si}||(t_i, c_{ij}, r_{ij})$.

1. $U_i \rightarrow U_j$: $R_{si}||t_i$.
2. U_j: verify $t_i = (m_i, \sigma_i)$ and all $t_k = (m_k, \sigma_k)$ in R_{si}, verify $\sigma_0 = (\mathbf{RO}||\mathcal{H}(t_0))$, check all $c_n = \mathcal{H}(t_{n+1})$, and verify all proof in R_{si}.
3. $U_i \leftrightarrow U_j$: $PK\{(u_i, w_i): (m_{i,1} = m_{i,2}^{u_i} \wedge m_{i,2} = g^{w_i})\}$.
   i. U_i: pick $(r_1, r_2) \in_R \mathbb{Z}_q^2$, computes $t_1 := m_{i,2}^{r_1}, t_2 := g^{r_2}$, send t_1, t_2 to U_j.
   ii. U_j: pick the challenge $c \in \mathbb{Z}_Q^*$, send c to U_i.
   iii. U_i: compute $s_1 := r_1 - cu_i, s_2 := r_2 - cw_i$, send s_1, s_2 to U_j.
   iv. U_j: accept if $m_{i,2}^{s_1} m_{i,1}^{c} = t_1, g^{s_2} m_{i,2}^{c} = t_2$ holds.
4. U_j: select a new tag token t_j and send $c_i = \mathcal{H}(t_j)$ to U_i.
5. $U_i \rightarrow U_j$: the response tag $r_i = (g^{c_i} h)^{u_i}$.
6. $U_i \leftrightarrow U_j$: $PK\{(u_i): (m_{i,1} = m_{i,2}^{u_i} \wedge r_i = (g^{c_i} h)^{u_i})\}$.
   i. U_i: pick $r \in_R \mathbb{Z}_Q$, compute $t_1 := m_{i,2}^{r}, t_2 := (g^{c_i} h)^{r}$, send t_1, t_2 to U_j.
   ii. U_j: pick the challenge $c \in \mathbb{Z}_Q^*$, send c to U_i.
   iii. U_i: compute $s := r - cu_i$, send s to U_j.
   iv. U_j: accept if $m_{i,2}^{s} m_{i,1}^{c} = t_1, (g^{c_i} h)^{s} r_i^{c} = t_2$ holds.
7. U_j: extend transfer chain $R_{sj} = R_{si}||(t_i, c_i, r_i), Proof_{ij}$

3.3.6 **Authentication**

After authentication, AP could start to provide service.

1. $U_i \rightarrow AP$: $R_{si}||t_i$.
2. AP: verify $t_i = (m_i, \sigma_i)$ and all $t_j = (m_j, \sigma_j)$ in R_{si} are never used, verify $\sigma_0 = (\mathbf{RO}||\mathcal{H}(t_0))$, check all $c_n = \mathcal{H}(t_{n+1})$, and verify all proof in R_{si}.

3. $U_i \leftrightarrow AP$: $PK\{(u_i, w_i): (m_{i,1} = m_{i,2}^{u_i} \wedge m_{i,2} = g^{w_i})\}$.

i. U_i: randomly pick $(r_1, r_2) \in_R \mathbb{Z}_Q$.

ii. U_i: compute the challenge $c \coloneqq \mathcal{H}(g||m_{i,1}||m_{i,2}||m_{i,2}^{r_1}||g^{r_2})$.

iii. U_i: set $s_1 \coloneqq r_1 - cu_i, s_2 \coloneqq r_2 - cw_i$, and send (c, s_1, s_2) to AP.

iv. AP: accept the proof if $c = \mathcal{H}(g||m_{i,1}||m_{i,2}||m_{i,2}^{s_1} m_{i,1}^{c}||g^{s_2} m_{i,2}^{c})$ holds.

4. AP: add (R_{si}, t_i) to **LOG**.

3.3.7 Trace

To trace the dishonest user U_i's idendity ID_i in two records R_{sj} and R_{sk} with the same prefix $\ldots||(t_i, c_{ij}, r_{ij})$ and $\ldots||(t_i, c_{ik}, r_{ik})$ in **LOG**.

1. AP: compute $\left(\frac{r_{ij}}{r_{ik}}\right)^{\frac{1}{c_{ij}-c_{ik}}} = \left(\frac{g^{c_{ij}}h}{g^{c_{ik}}h}\right)^{\frac{u_i}{c_{ij}-c_{ik}}} = g^{u_i} = ID_i$.

3.4 System Implementation

3.4.1 Develop environment

Out system is implemented on Microsoft .Net framework and use the FMOD Ex sound system [31] as the audio engine. We implement our scheme to provide an online music service DRM system. We adopt SHA1 hash function and Triple-DES encryption in our system implementation.

3.4.2 System Architecture

Figure 3 shows the overview of our system implementation. The server module includes the register server and the content server. The client player first registers his ID first and then anonymously proves his identity to get new tag tokens. The client player shows a new tag token to the content server to download some protected contents. The player can directly

request the license to play the content or transfer the content to another client player.

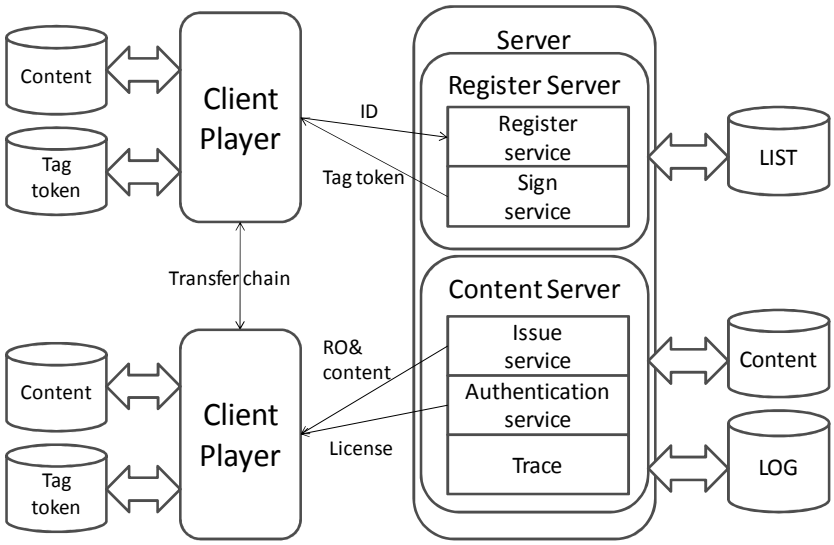


Figure 3 Overview of our system implementation

☐ The main functionalities of the server module

1. Register server provides Register and Sign service and maintains the public identification **LIST**. If a user is traced in the Trace algorithm by content server, the user will be revoked.
2. The content server provides Issue and Authentication service and maintains the authentication **LOG**.
3. The registered user can show his tag token to access the Issue and Authentication service, and the revoked users are not allowed to access the Sign service to get any new tag token.
4. The content server can import music contents and set the Rights of the contents, which includes the times of playing the content, the times of transferring the content, the expiration time after first play, the expiration time after getting the content (see Figure 4).

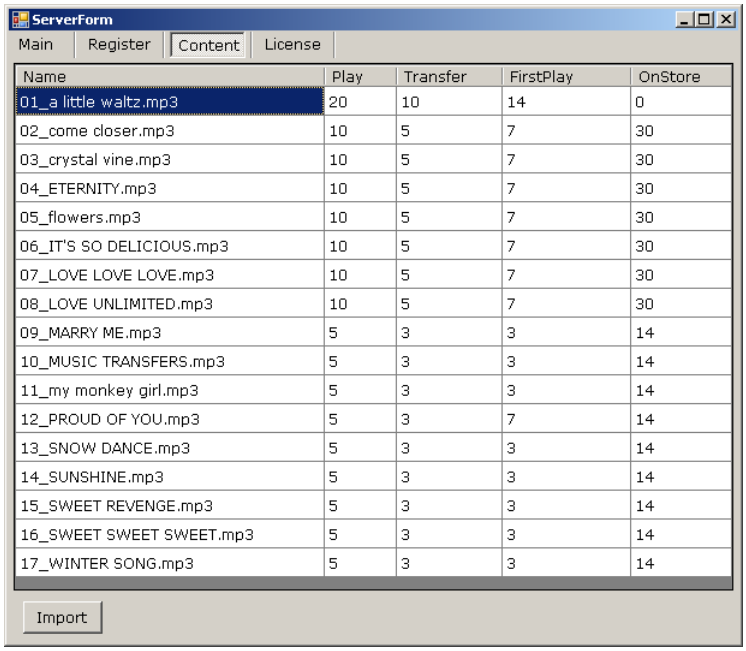


Figure 4 Setting the Rights

5. Users can connect to the content server to download the protected content.
6. If the user can pass the authentication, the content server will send the corresponding license to the user.

☐ The main functionalities of the client player

1. The client player can connect to the content server to get the content list, download the content in the list, and add into a local playlist (see Figure 5).

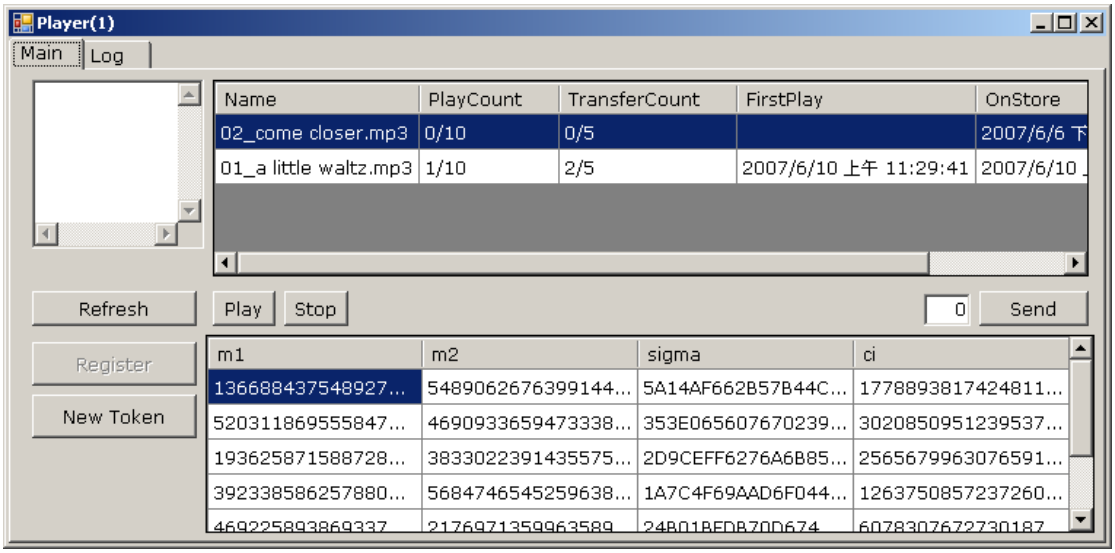


Figure 5 Playlist and Client Player

2. The client player can transfer its content to another client and remove the content file from its playlist.
3. Without illegal backup and restore, the client player cannot transfer the same protected content to multi clients.

3.4.3 **Discussion**

We discuss some issues in implementation.

☐ Issue

In our original scheme, we set the bound *k* equal to 1 to make sure there is only one user access the service simultaneously. For lager *k*, the user can send multi hash value of tag token to the **AP** in the **Issue** phase, and the receiver still commit only one hash value in the **Transfer** phase. It makes possible that there are *k* valid users to access the service for the same license.

☐ Transfer

Our system can support various payment systems with some modification. For the purchased model, the user can pay for new tag token which has two types "to download" and "to transfer". The user could use the "to download" tag token to download protected contented in the **Issue** phase and apply the "to transfer" tag token to get new contents from the other user.

☐ Authentication

If the license is expired, the user needs to re-authenticate to **AP**. Because the default bound *k* is set to 1, **AP** doesn't need to send challenge in the **Authentication** phase and it makes re-authenticate easilier. But if we set the bound *k* more than one and modify the **Authentication** protocol that send the challenge to the user, the user need to transfer the content to himself and extend the authorization transfer chain to avoid his identity to be traced. We can only reject the dishonest user if the length of his authorization transfer chain is shorter

than the record in authentication **LOG** in the original scheme, and there will be another advantage that we can trace out the identity of dishonest user after modification.

- Trace

The **Trace** algorithm can find the identity of dishonest user who first transferred to multi parties. By considering of the computing and storage overhead, we can try to match all records in authentication **LOG** to find all dishonest users, and it will be a trade-off between performance and security.

Chapter 4

Security Analysis

In this chapter, we first present the standard definition of Diffie-Hellman problem and give both decisional and computational Diffie-Hellman assumption in section 4.1. Then we give the definition of the properties of an anonymous and authorization transferable DRM system in section 4.2 and give the security proof of our scheme in section 4.3.

4.1 Security Assumptions

Notation: Let p be a large prime number such that the discrete logarithm problem defined in $\mathbb{Z}_p^*$ is hard. Let $\mathbb{G} \in \mathbb{Z}_p^*$ be a cyclic group of prime order q and g is a random generator of $\mathbb{G}$. a and b are two elements randomly chosen from $\mathbb{Z}_q - \{0\}$.

4.1.1 Computational Diffie-Hellman (CDH) assumption

The Computational Diffie-Hellman (CDH) problem [10] can be defined as the task of computing g^{ab} given g, g^a, and g^b. The CDH assumption is: for all probabilistic polynomial time adversaries Adv,

$$\Pr\left[a, b \xleftarrow{R} \mathbb{Z}_q^2 : g^{ab} \leftarrow \mathrm{Adv}\left(\mathbb{G}, g, g^a, g^b\right)\right] < \frac{1}{\mathrm{poly}(k)}$$

4.1.2 Decision Diffie-Hellman (DDH) assumption

The Decision Diffie-Hellman (DDH) problem [3] is defined as the task to distinguish g^{ab} from a random element in $\mathbb{G}$, given g, g^a, and g^b. The DDH assumption is: for all probabilistic polynomial time adversaries Adv,

$$\Pr\left[\begin{array}{l} a, b \xleftarrow{R} \mathbb{Z}_q^2; x_0 = g^{ab}; x_1 \xleftarrow{R} \mathbb{G}; \\ d \xleftarrow{R} \{0,1\}; d' \leftarrow \mathrm{Adv}\left(\mathbb{G}, g, g^a, g^b, x_d\right) \end{array} : d = d'\right] < \frac{1}{2} + \frac{1}{\mathrm{poly}(k)}$$

4.1.3 **Strong RSA Assumption**

The Strong RSA Problem is given an RSA modulus n and a random element $z \in \mathbb{Z}_n^*$, to find a pair $(u, e) \in \mathbb{Z}_n^* \times \mathbb{Z}$ such that $u^e = z$ and $e > 1$.

For all probabilistic polynomial time adversaries Adv,

$$\Pr\left[z = u^e \wedge e > 1 : z \xleftarrow{R} \mathbb{Z}_n^* \ (u, e) \coloneqq \mathrm{Adv}(n, z)\right] < \frac{1}{\mathrm{poly}(k)}$$

4.2 Security Definition

Our system consists of three usual players: the register server S, the content server AP, and the user U_i; together with two algorithm: **Setup** and **Trace**, and five protocol: **Register**, **Sign**, **Issue**, **Transfer**, **Authentication**, and a unforgeable signature scheme Σ.

☐ Definition

1. The **Setup**(1^k) algorithm is a key generation algorithm run by S and AP. It takes the security parameter 1^k as input. The algorithm outputs the key pairs (sk_s, vk_s) and (sk_{AP}, vk_{AP}).
2. In the **Register**$(S(pk_{U_i}), U_i(sk_{U_i}))$ protocol, the user U_i register his public identity ID_i to the register server S. The register server S maintain a public identification **LIST** and adds a new record (i, ID_i).
3. In the **Sign** $\left(S(sk_S, \mathbf{LIST}), U_i\left(sk_{U_i}, \mathbf{LIST}\right)\right)$ protocol, the user U_i proves he is one of the register users in **LIST** to the register server S and get a tag token t_i.
4. In the Issue $\left(AP(sk_{AP}, vk_S), U_i\left(sk_{U_i}, t_i\right)\right)$ protocol, the user U_i shows a fresh tag token t_i to the content server AP to exchange a protected content and corresponding Right Object **RO**. AP obtains a proof π of validity of the token. U_i initials a transfer chain R_{si}.
5. In the **Transfer** $\left(U_i\left(sk_{U_i}, t_i, R_{Si}\right), U_j\left(vk_S, vk_{AP}, sk_{U_j}, t_j\right)\right)$ protocol, the user U_i shows

transfer chain R_{si} and tag token t_i to the other user U_j. U_j obtains a proof π' of validity of the chain and token and a extended transfer chain R_{sj}.

6. In the **Authentication** $\left(U_i(sk_{U_i}, t_i, R_{Si}), AP(vk_S, vk_{AP})\right)$ protocol, the user U_i shows the transfer chain R_{si} and tag token t_i to the content server AP. AP obtains a proof π'' of validity of the chain and token and adds (R_{si}, t_i) to **LOG**. U_i updates the corresponding **RO**.
7. The **Trace**$(t_i, R_{sj}, R_{sk}, \mathbf{LOG}, \mathbf{LIST})$ algorithm run by anyone outputs the dishonest user's identity. It inputs the authentication **LOG** with two records including transfer chains R_{sj} and R_{sk} belongs to the same tag token t_i, and the register users **LIST**. The algorithm outputs the identity ID_i of the dishonest user U_i.

☐ Security Property

We will now define the security properties for our DRM system

- **Correctness**.

  If an honest user runs **Register** and **Sign** with an honest register server and runs **Issue** with an honest content server, he will get a valid tag token and an initial transfer chain. If all users in the transfer honestly follow **Transfer** protocol, when the last user runs **Authentication** with an honest content server, the content will accept the token and transfer chain with an overwhelming probability.

- **Unforgeability**.

  The adversary who has not registered his identity to S cannot generate a tag token. Further, the adversary who doesn't know secret to some token cannot successfully run **Transfer** nor **Authentication** with an honest user or AP by showing this token.

- **Anonymity**.

  When U_i runs **Authentication** with AP, AP cannot learn anything about the identity of U_i from the tag token and transfer chain.

- **Traceability**.

  Suppose U_j and U_k are honest users. If U_i runs **Transfer** respectively to U_j and U_k with the same tag token t_i and the corresponding **RO**. The receivers' output are (R_{sj}, t_j) and (R_{sk}, t_k). Then the **Trace**$(t_i, R_{sj}, R_{sk}, \mathbf{LOG}, \mathbf{LIST})$ algorithm will output ID_i with high probability.

We prove our system with all the properties described under the Computational Diffie-Hellman assumption and the Decision Diffie-Hellman assumption.

4.3 Security Proof

Theorem. *The purposed DRM system has the properties of correctness, unforgeability, anonymity, and Traceability under DDH assumption and Strong RSA assumption in the random oracle model.*

Proof:

☐ Correctness

The correctness is easily observable.

☐ Unforgeability

The form of a tag token $t_i = (m_i, \sigma_i) = \left(m_i, \mathrm{Sign}_{SK_S}(m_i)\right)$, the tag token's unforgeability is from unforgeability of the underlying signatures scheme.

In **Issue**, **Transfer**, and **Authentication** protocol, adversary A is required to prove $\mathrm{PK}\{(u_i, w_i): (m_{i,1} = m_{i,2}^{u_i} \wedge m_{i,2} = g^{w_i})\}$ with some tag token t_i. The NIZK proof (c, s_1, s_2) is a signature of knowledge (u_i, w_i), and the unforgeability is already proved in [7].

☐ Anonymity

Under the Strong RSA assumption, in the **Issue**, **Transfer**, **Authentication** protocol,

the user shows a zero-knowledge proof of knowledge of a tag token and a corresponding tag token secret key.

$$\mathrm{PK}\{(u_i, w_i): (m_{i,1} = m_{i,2}^{u_i} \wedge m_{i,2} = g^{w_i})\}$$

Assuming that the hash function $\mathcal{H}$ is a random function, the NIZK proof (c, s_1, s_2) do statistically not reveal any knowledge. Hence, deciding whether a tag token $(c, s_1, s_2, m_{i,1}, m_{i,2})$ originates from some group member requires to compute the discrete logarithm of $m_{i,1}$ to the base ID_i. It will violate the discrete logarithm assumption.

Linking two tag token, i.e., deciding whether two tag token $(c, s_1, s_2, m_{i,1}, m_{i,2})$ and $(c', s_1', s_2', m_{i,1}', m_{i,2}')$ originate from the same registered user requires to decide whether $\log_{ID_i} \frac{m_{i,1}}{m_{i,1}'} = \log_g \frac{m_{i,2}}{m_{i,2}'}$ with some ID_i. As (c, s_1, s_2) and (c', s_1', s_2') do not reveal any useful knowledge. It will violate the DDH assumption.

☐ Traceability

If a dishonest user U_i runs **Transfer** to U_j and U_k with the same tag token t_i. U_j and U_k get new transfer chain and tag token (R_{sj}, t_j) and (R_{sk}, t_k). After both of them run **Authentication** with AP, the $\mathrm{Trace}(t_i, R_{sj}, R_{sk}, \mathbf{LOG}, \mathbf{LIST})$ algorithm will output ID_i.

$$\left(\frac{r_{ij}}{r_{ik}}\right)^{\frac{1}{c_{ij}-c_{ik}}} = \left(\frac{g^{c_{ij}}h}{g^{c_{ik}}h}\right)^{\frac{u_i}{c_{ij}-c_{ik}}} = g^{u_i} = ID_i$$

We have proved correctness, transferability, unforgeability, and anonymity and complete the proof. ☐

Chapter 5

Conclusion and Future Work

5.1 Conclusion

In this thesis, we have purposed a secure DRM system with anonymity and transferability. It has the properties of correctness, unforgeability, anonymity, transferability, and traceability. We showed the reason why anonymity and transferability are important and are not supported in current DRM systems. In our system, the user can obtain anonymity and keep personal's privacy. The user can also freely transfer his purchased content and the authorization to his family or friends. On the other hand, the server side doesn't worry about the mass distribution. For each purchased content, there is only one legal copy allowed to refresh its license, and it is easy for server to trace the illegal user's identity.

Our system can support various payment systems. For the subscription based model, the user can ask unlimited tag tokens to download an unlimited amount of contents during subscription, and the DRM system set the license valid according to the corresponding token. For the purchased model, the user can pay for new tag token which has two types "to download" and "to transfer". The user could use the "to download" tag token to download protected contented and apply the "to transfer" tag token to get new contents from the other user.

In Our scheme, the honest user's privacy is protected as much as possible. The server cannot know who has bought the content nor monitor the user's activity history or statistic, but the server can still gather their product statistics for market research.

5.2 Future works

In our scheme, the transfer chain size is linear to the numbers of transfer. It adds the overhead of the verifier and the server to maintain the **LOG**. Because we trace the illegal user

by the transfer order in the transfer chain, some improvement discussed in [17][20][22] for the original *k*-TAA scheme can not be applied to our scheme. We need some method to reduce the computing and storage overheads.

Next, we don't provide a secure offline trial mechanism in our scheme. The user is able to offline-pass the protected content and corresponding license to the next user, and the receiver is allowed to play the file by the license directly. The receiver may modify system time and backup the license to escape the limitation of some rights like PlayCount or FirstPlay. However, it is required extra mechanisms (secure clock, etc.) to prevent the backup-restore attack for a DRM system to provide an offline trial mechanism.

The trend of the market is opposed to DRM, but it is impossible to expect all companies open their products to DRM-free. The future DRM systems need to satisfy both consumers' and providers' demands to reach the balance between the rights of the providers and the accessibility of the public. It is also required an open standard for the compatibility to enable protection across platforms and systems.

Bibliography

[1] Arnab, A. Hutchison. Digital Rights Management - A current review. Departmental technical report, no. cs04-04-00, University of Cape Town, 2004.

[2] G. Ateniese, J. Camenisch, M. Joye, G. Tsudik. A Practical and Provably Secure Coalition-Resistant Group Signature Scheme. In *CRYPTO 2000*, LNCS 1880, pp. 255-270, Springer-Verlag, 2000.

[3] D. Boneh. The decision Diffie-Hellman problem. In *Algorithmic Number Theory* (*ANTS-III*), vol. 1423 of LNCS, pp. 48–63, Springer-Verlag, 1998.

[4] D. Boneh, X. Boyen, H. Shacham. Short group signatures. In *CRYPTO 2004*, LNCS 3152, pp. 41–55, Springer Verlag, 2004.

[5] J. Camenisch, M. Michels. A group signature scheme based on an RSA-variant. Tech. Rep. RS-98-27, BRICS, Dept. of Comp. Sci., University of Aarhus, preliminary version in *Advances in Cryptology | ASIACRYPT '98*, vol. 1514 of LNCS.

[6] D. Chaum, E. van Heyst. Group signatures. In *Eurocrypt 1991*, volume 547 of LNCS, pages 257–65. Springer-Verlag, 1991.

[7] D. Chaum, T. P. Pedersen, Transferable Cash Grows in Size. In *Advances in Cryptology – EUROCRYPT' 92*, pp. 390-407, 1992.

[8] D. Chaum, T. P. Pedersen. Wallet databases with observers. In *Advances in Cryptology - CRYPTO '92*, volume 740 of LNCS, pp. 89-105. Springer-Verlag, 1993.

[9] R. Cramer, I. Damgård, B. Schoenmakers. Proofs of Partial Knowledge and Simplified Design of Witness Hiding Protocols. In *Advances in Cryptology – CRYPTO '94*, Vol. 839 of LNCS, pp 174-187. Springer-Verlag, 1994.

[10] W. Diffie, M. E Hellman. New directions in cryptography. In *IEEE Trans. on Information Theory*, IT-22(6):644–654, November 1976.

[11] J. Feigenbaum, M. Freedman, T. Sander, A. Shostack. Privacy Engineering for Digital Rights Management Systems. In *Proceedings of the 2001 ACM Workshop on Security and Privacy in Digital Rights Management*. 2002.

[12] R. Grimın. Privacy for Digital Rights Management Products and their Business Cases. In *Innovationspotenziale der Informationstechnik 2005*.

[13] T. C. Lam, V. K. Wei, Mobile Agent Clone Detection using General Transferable E-Cash. In *InfoSecu'2002*, 2002.

[14] Q. Liu, R. Safavi-Naini, N. P. Sheppard. Digital Rights Management for Content Distribution. Australasian Information Security Workshop (AIS W), Adelaide, South Australia, 2003.

[15] S. Michiels, K. Verslype, W. Joosen, B. D. Decker. Towards a Software Architecture for DRM. In *Proceedings of the 5th ACM Workshop on Digital Rights Management*, 2005

[16] D. K. Mulligan, J. Han, A. J. Burstein. How DRM-Based Content Delivery Systems Disrupt Expectations of "Personal Use". In *Proceedings of ACM Workshop Digital Rights Management 2003*, pp. 77-89.

[17] L. Nguyen, R. Safavi-Naini. Dynamic *k*-Times Anonymous Authentication. In *ACNS 2005*, LNCS 3531, pp. 318–333, 2005.

[18] A de Santis, L. di Crescenzo, G. Persiano, M. Yung. On Monotone Formula Closure of SZK. In *35th FOCS*, IEEE, pp. 454-465, 1994.

[19] P. Schnorr. Efficient signature generation for smart cards. Journal of Cryptology, 4(3):239-252, 1991.

[20] R. Shigetomi, A. Otsuka, J. Furukawa, K. Martin, H. Imai. A Provably Secure Refreshable Partially Anonymous Token and Its Applications. The Institute of Electronics, Information and Communication Engineers, 2006

[21] Teranishi, J. Furukawa, K. Sako. *k*-Times Anonymous Authentication. In *ASIACRYPT 2004*, LNCS, pp. 308–322, Springer, 2004.

[22] Teranishi, K. Sako. *k*-Times Anonymous Authentication with a Constant Proving Cost. In *PKC 2006*, LNCS 3958, pp. 525–542, 2006.

Web Resource

[23] Adobe Acrobat

http://www.adobe.com/products/acrobat/

[24] Apple iTunes Store

http://www.apple.com/itunes/

http://www.apple.com/itunes/store/

http://www.apple.com/legal/terms/site.html

[25] Apple announced 100 Million iPods Sold.

http://www.apple.com/pr/library/2007/04/09ipod.html

[26] Apple Launches iTunes Plus

http://www.apple.com/pr/library/2007/05/30itunesplus.html

[27] Copyright Act.

http://www.tipo.gov.tw/eng/laws/e1-4-1an95.asp

[28] Digital Millennium Copyright Act.

http://www.copyright.gov/legislation/dmca.pdf

[29] eXtensible rights Markup Language (XrML) 2.0 Specification,

http://www.xrml.org/

[30] ezPeer+

http://web.ezpeer.com/index.php

[31] FMOD sound system

http://www.fmod.org/

[32] KKBOX

http://www.kkbox.com.tw/

[33] Napster music store.

http://free.napster.com/

[34] Wal-Mart Music Downloads

http://www.walmart.com/music_downloads/introToServices.do

[35] Windows Media Rights Manager

http://www.microsoft.com/windows/windowsmedia/forpros/drm/

[36] Yahoo! Music.

http://www.musicmatch.com/