Towards Privacy Preserving Digital Rights Management Using Oblivious Transfer

Hung-Min Sun[1], King-Hang Wang[2], and Chi-Fu Hung[3]
Department of Computer Science, National Tsing Hua University
[1]hmsun@cs.nthu.edu.tw, {[2]khwang0, [3]duck4011}@is.cs.nthu.edu.tw

Abstract

Digital Rights Management (DRM) is a hot topic in digital content development. Many implementations invade users' privacy by revealing what contents they have purchased. Preserving user's privacy during purchasing content is necessary without doubt.

Some works address the problem by providing anonymity to the user. Anonymous trade would not allow the shopkeepers to manage his customers efficiently. Also, the identities of users can still be profiled via side channels like routing paths or IP addresses. In this paper, we propose a scheme to preserve users' privacy by hiding the users' choices of contents from the shopkeepers using oblivious transfer (OT). We will evaluate our scheme in the aspects of security, performance, comparison, and implementation. A privacy measurement called "Privacity" is also firstly defined.

Index terms: Digital Rights Management, DRM, Privacy, Oblivious Transfer, Electronic Commerce

I. Introduction

Digital Rights Management (DRM) systems [12], [16] are designed to protect and manage digital contents. They not only protect the intellectual property, but also maintain the revenue of the content providers. Consequently illegal copy of digital content will be reduced. A DRM system allows authorized users to access media contents under the conditions stated in the licenses. Unauthorized users cannot access the contents.

There are many DRM systems used in the business world. Microsoft and Apple are the famous vendors of DRM systems. Microsoft Media Right Manager system [18] is implemented on the Microsoft Media Player platform. iTunes [1] is supported by Apple. Users can use it to search and download digital music. The downloaded music files are under the control of the iTunes system.

Along with the installation of DRM enforcement software on PC, privacy issue is concerned [3], [9], [10], [14], [20]. Usually, the software will collect information, like types of content playing, playing frequency, or duration, from users' computers. In addition, for billing purpose, users' personally identifying information (PII), like name, social security number, address, credit card number, etc. will be transferred to the content provider. Problems like excess information collection or information misuse not only discourage consumers in using DRM enabled technology, but also raise legal disputes between users and content providers. Feigenbaum *et al.* [10] listed some principles for privacy engineering on DRM. These principles include collection limitation, data accuracy, purpose disclosure, use limits, security, openness, participation, and organizational accountability. Content provider should obey these principles in order to protect consumers' rights.

Let us focus on the issue collection limitation - which restricts the content provider to collect the minimum set of information that he needs. Consider the traditional call-and-delivery business, like pizza delivery. The shopkeeper has to know at least what the consumer buys, where he lives, and perhaps the credit card number for billing purposes. Other sensitive information like customers' name or social security number may be excluded from the shopkeeper database if this information is not affecting business. In a digital business, like internet shop, the shopkeeper should keep a similar manner in treating users' PII – minimize the information collection on user's related data. Currently, there is no technological enforcement on information collection proposed in DRM system. User privacy can only be protected by mutual agreement [29] or local laws [6][22]. In this paper, we wish to provide this enforcement via a technology path.

Furthermore, if we can preserve the user privacy by concealing the contents they purchase, it will bring benefit to the content provider and the consumers. This means the content provider should not collect the information about what the user have purchased and stored in the computer using DRM enforcement software, and also more importantly, with the help of the transaction protocol, content provider should not be able to know what the consumer is purchasing from him. We will discuss this more thoughtfully in the next section, including the reasons and advantages in implementing this. But to illustrate the basic concept of the concealment, we show an example here. Consider a shop with only one exit, in where the casher is. Customer enters the shop with an empty black plastic bag. He puts the things he wants to buy into the bag and weight the bag in the casher. By measuring the weight of the bag, the shopkeeper knows how much is the customers have to pay. This business works only if the ratios of weight-to-price are the same for all the products. This business is quite infeasible, but it preserves the customers' *privacy during purchase* (PDP). We refer those business preserves customers' PDP as private trade.

A few literature [11][15][24] concerned this problem and propose "anonymous trade" to preserve customers' PDP. That means the customers are not necessary to provide their identity during purchase. Although the servers cannot know the relationship between the users and the contents, anonymous trade would not allow the shopkeepers to manage users efficiently, like giving discount to regular buyers. Additionally the identities of users may still be profiled via side channels like routing paths or IP addresses. Furthermore, anonymous trade can only be applied to the e-cash payment system [27] and it is inconvenient and risky for consumers (using credit card will immediately expose the identity of the users).

We purpose a protocol for private trade by using oblivious transfer (OT) scheme. The shopkeeper will only know how many things a user buys, however, exactly what items a user buys, the shopkeeper will have no idea. Private trade allows shopkeeper can efficiently manage his customers and employ different payment schemes. There are several advantages in using private trade and will be discussed in a more detail fashion later.

The rest of this paper is organized as follows. In Section II, we provide a strong argument on why should we preserve customers' PDP. In section III, we describe the oblivious transfer (OT) schemes proposed by Chu *et al.*, and review the privacy issue discussed in previous literature. In Section IV, we present our approach in acquiring licenses and updating licenses. Next, we focus on the secure issue of our approach in Section V, and in Section VI we compare our approach with other well-known schemes. In Section VII, the issue of implementation will be raised. Finally, a conclusion will be drawn in Section VIII.

II. Issue for preserving customers' PDP

Privacy engineering on DRM should be discussed based on the following four aspects: customers' need, economic aspect, technical aspect, and legal aspect. We will explore preserve customers' PDP regarding these aspects.

A. Customers' Need

Traditional retail shops can be divided into membership and non-membership types. These two types of business benefit customers in different aspects. Non-membership type usually provides PDP to customers, that is, the customers' related information is not linked to what they are purchasing. Membership type allows regular customers to receive refunds and latest catalogs from the retail shops. Of course, from the view of customers, it is very ideal to have both advantages. Yet, no business in the physical world provides both benefits to customers. But at least, customers can choose.

For e-commerce, also for digital content business, customers also concern about privacy and convenience. Currently, there are many membership type DRM solutions provided in the market and works well [1][8][13][18]. The acceptance for non-membership internet business is low for both customers and content provider. Previous works [4][11][15][24] provide anonymity to users to preserve their PDP. From customers' point of view, it is reasonable to enjoy PDP in traditional retails shop, as well as in e-commerce. It may be a little "out-of-scope", the private trade proposed in this paper may provide both PDP and convenience. We shall explore that in section III. What we want to conclude here is customers should have the right to choose to enable their PDP.

B. Economic Aspect

Feigenbaum *et al.* [10] founded a core stone in privacy engineering of DRM. They thought from economic aspect there is less incentive to preserve user privacy. It is without doubt that privacy-enabled DRM will cost more money. And more important, knowing better the customer can help in managing risk, customization, billing, and setting prices for products. Customers' privacy is not interested by content providers. Preserving privacy is simply because of laws.

However, we would like to point out that preserving PDP will benefit content provider as well. In some conservative countries, like countries in the eastern Asian, people are not open to purchase sexual related product, like condom or adult movie. With the help of the internet, people are much more willing to buy. Apart from the convenience of e-commerce, less embarrassing is believed as the major factor for pushing the business. Preserving customers' PDP will definitely encourage them in purchasing sexual related product. This immediately implies the content provider would make more money. Similar concept can be applied to any other privacy-sensitive products like medical, or slim up products.

C. Technical Aspect

If the implementation of preserving customers' PDP is infeasible, or cost too much, or totally unsecured, we may only consider PDP protection in philosophical level. Previous literature [4][11][15][24] provided some solutions on preserving customers' PDP using anonymous trade. Their solutions do not revolute the DRM infrastructure or cost unreasonable overhead packets. In this paper, we propose private trade which provides fully secured customers' PDP. Also, the cost of our implementation is limited and acceptable.

D. Legal Aspect

The most famous legislation concerning privacy is the Directive 95/46/EC of the European Parliament and the Council of 24 October 1995, or referred as the Directive [22]. The Directive proposed several principles [14] on processing individual information. Of course, preserving customers' PDP does not conflict with the principles. And if customers are not willing to hand-in his purchase information to the content provider, according to the *Rights* [14], he can "raise certain objections regarding the controller's execution of these principles". Here we have the legal foundation for preserving PDP.

From the four aspects, we can see that preserving customers' PDP is necessary.

III. Related Works

A. Oblivious Transfer

Oblivious transfer (OT) is a cryptographic primitive developed since the first scheme by Rabin [25]. And it is well developed since then [2], [7], [19], [21], [23], [28]. A receiver *R* is allowed to obtain some information from a sender *S* via some interactions in OT. A secure oblivious transfer demands that a receiver *R* can obtain some information from a sender *S* in which 1) *R* can only get what he requested for. 2) *S* has no idea of which information *R* gets. A specific OT is called *k*-out-of-*n* OT scheme. This scheme specifies that *S* has *n* messages and *R* wants to get *k* of them through a secure OT scheme. A k-out-of-n OT scheme is denoted as OT_n^k.

B. DRM Privacy Issue

Most of the vendors of DRM system do not provide the privacy protection. Microsoft Media Right Manager [18], IBM EMMS [8], InterTrust's DRM system [13], and Apple iTunes [1] are all the famous DRM system recently, but none of them protects the users' privacy. In order to authenticate with the users and charge correctly, privacy may be an optional function in these vendor DRM systems.

"Concealing the user's identity or public key" is the most popular way to protect privacy in academic curriculum. The license server cannot recognize the user because the user's ID is unknown. There are many methods to realize such a scheme. Both Park *et al.*'s privacy enhancing protocol [24] and Grimm *et al.*'s approach [11] use temporal identity (or called pseudonymous consumer ID) to replace the user's ID. Lee *et al.*'s scheme [15] applies the properties of electronic cash to provide the anonymity. Moreover, Conrado *et al.*'s method [4] protects the user's public key by applying hash function.

All protection schemes mentioned above are indeed providing anonymity. They hide the user's identity or other significant information, so the license server cannot recognize the user. However, it is not suitable to solve the problem by employing anonymity as we have discussed in the introduction part. Therefore, we present a new way to handle the privacy issue by private trade. We make sure the license server cannot know the relations between the users and the contents. This means the license server knows the user's identity, but they do not know what the user buys.

The privacy of private trade is based on the how many contents does the license server sell and how many contents does a customer buy. Assume that there are *n* contents in the license server and the customer wants to purchase *k* of them. Obviously, if *n* is smaller, the choices of the customer are confined, and it is easier to "guess" what the customer buys. For the same reason, if *k* is larger, it is easier to "guess" what a customer wants. In this paper, we also define the term *Privacity* which measures the degree of privacy for a customer has.

IV. The Proposed Scheme

We propose a scheme to realize privacy protection in this section. The scheme consists of three phases: *Register phase*, *License acquirement phase* and *License update phase*. This is realized by applying Chu and Tzeng's OT scheme. Before illustrating our protocols, we discuss the environment of our DRM system.

The new scheme can be applied to any DRM system which can transfer the license and the content separately, such as Microsoft Media Rights Manager system (one of the device-based DRM system) or Sun *et al.*'s identity-based DRM system [26]. It is an easy way to apply our application. Only you should do is to replace the old *License acquirement* and *License Update* protocol with the new one of our application, respectively. Moreover, the remainders of the protocol of the DRM system such as *Content Requesting* and *Content Rendering* etc. do not need to change. Practically, DRM systems which only provide combined-license delivery (that combines the license with the content) are not cost-effective for this application. For offline DRM system, we can omit the license update phase.

A. Register Phase

All users who want to acquire license by our application should register first. Before the user acquires the corresponding license, the user should register to the license server to be a member. Once the user becomes a member of the license sever, the user will have a *User ID* for license acquirement and, moreover, the user should provide bank account or contacting information for payment.

B. License Acquirement Phase

The *License Acquirement* protocol is activated when a user is going to pay and get the corresponding license after the content is downloaded. We illustrate the *License Acquirement* protocol in Figure 1 below.

1) Calculating the OT factors and sending request message to the server: when the user has downloaded the content and prepare to acquire the corresponding license, this phase will occur. The user can retrieve the *License ID* of the content from the web. Then, the user calculates *OT factors* according to this *License ID*. (The *OT factors* will be discussed in detail later.)

After calculating the *OT factors*, the user sends it with his *User ID*, the *content price*, and the signature of this message to the license server. This signature is created by the user's private key, and this private key can be stored in the smart card for identity-based DRM system or in the personal computer for device-based DRM system.

2) Calculating the OT result and sending response message to the user: the license server verifies the *Content price* and the amount of contents of the user purchases. Then, the server calculates the corresponding *OT result* with the received *OT factors*. This *OT result* can be digested by the user to get the correct license, but the license server does not know which license in this group the user requests. In addition, the user is unable to retrieve more licenses that he purchases. (The *OT result* will be discussed in detail later, too.)

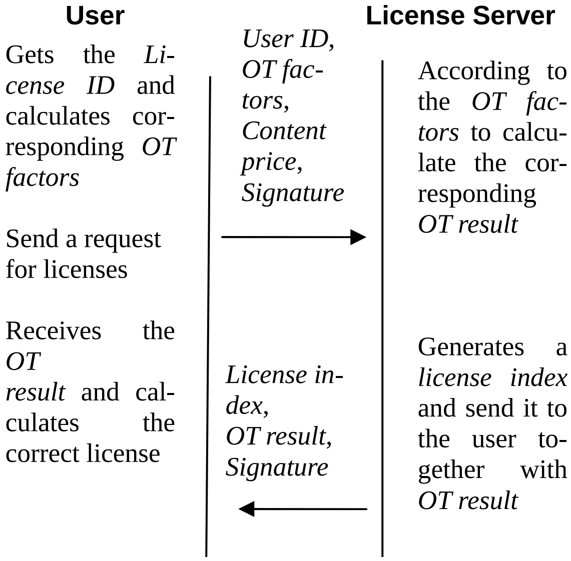
User
License Server
Gets the License ID and calculates corresponding OT factors
Send a request for licenses
Receives the OT result and calculates the correct license
User ID, OT factors, Content price, Signature
License index, OT result, Signature
According to the OT factors to calculate the corresponding OT result
Generates a license index and send it to the user together with OT result

Fig 1. Protocol for License Acquirement

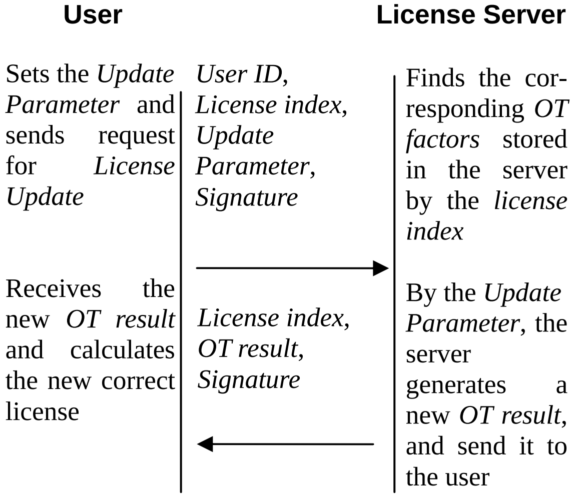
User
License Server
Sets the Update Parameter and sends request for License Update
Receives the new OT result and calculates the new correct license
User ID, License index, Update Parameter, Signature
License index, OT result, Signature
Finds the corresponding OT factors stored in the server by the license index
By the Update Parameter, the server generates a new OT result, and send it to the user

Fig 2. Protocol for License Update

Afterward the license server generates a license index. This license index is a random sequence number, and it will be stored together with the *OT factors* in the license server for the *License Update* protocol. The server consequently sends the license index, *OT result*, and the signature of this message to the user.

3) Getting the correct license: after all, the user can calculate the correct license by using the received *OT result*. The remainder actions of the DRM system such as *Content Rendering* are the same with the original protocol in each DRM system.

C. License Update Phase

In most DRM systems, the *Rights* will be stored in the license to describe the rules about content usage. These rights are expressed by *Rights Express Language* (REL). The license agent (LA) which controls and enforces the rights in the user's device will access the contents under the situation described in the corresponding rights. Sometimes the rights will change [5]. For example, the rights record the number of allowed access times, and once the user accesses the content, the record should subtract one. In other case, the rights can restrict the period of the content access. If the user wants to extend the periods, the change of the rights will happen. Then, the user will get another new license.

Therefore, this protocol will be activated when the situations like that we mentioned above happen. Before accessing the content, the license agent will check if the rights need to be updated. If the rights need to be updated after accessing, the *License Update* protocol will occur consequently. We take the "Number of allowed access times" for example and illustrate the *License Update* protocol in Figure 2.

1) *Setting the Update Parameter and sending request for License Update*: the license agent detects that the rights need to be updated and then sets the *Update Parameter*. The *Update Parameter* is a message to indicate what the new rights record. Then, the user sends the request for *License Update*. The message consists of *User ID*, license index, *Update Parameter*, and the signature of this message.

For example, the original rights restrict that "the user can access the content only three times". Then, before accessing the content, the license agent will detect it and set the *Update Parameter* correctly. The *Update Parameter* here will indicate that "the user can access the content only twice". After the license server receives it, server can revise the rights and generate the new license with this *Update Parameter*.

2) *Generating new OT result and sending response message to the user*: by using the license index, the license server finds the corresponding *OT factors* stored in the server before. Moreover, the server utilizes the received *Update Parameter* to revise the rights and generate new *OT result*. Consequently, the server sends the message which consists of license index, *OT result*, and the signature of this message to the user. We stress that the license index can only help the license server to re-

trieve the *OT factors* from its own database. It does not contain any information about the content or the *License ID*.

3) *Getting the correct new license*: according to the license index, the license agent can recognize if this new *OT result* is corresponding with which content. The user calculates the new license with the new *OT result*, and then replaces the old license with the new one. Finally, the license is updated successfully.

To avoid users disconnecting from the internet immediately after rending the content, we should update the license each time before playing the content. This can prevent dishonest users from keeping the same license to render content.

D. OT Factor and OT Result

In this section, we will illustrate the construction of *OT factors* in the way that the server will not know what the user wants to buy. The basic scheme will be first introduced and is followed by some other features, like license server having more contents or contents having different prices. The concrete *OT* scheme shown below is the work of Chu and Tzeng [7].

When a user wants to purchase *k* digital contents from the internet, he will download the contents from the content server and connect to the license server for the licenses. The user will obtain the *License IDs*, $C=\{C_{\sigma_1},C_{\sigma_2},...,C_{\sigma_k} \mid 1\le \sigma_i \le n\}$, and prices of these contents. For simplicity, we consider all contents have the same price. Suppose the license server has totally *n* different licenses for each content and these license set is $I=\{C_1,C_2,...,C_n\}$. The user will calculate the *OT factors* with $\{A_i = w_{\sigma_i} g^{a_i} \mid 1\le i\le k\}$. These *OT factors* will be sent to the license server.

Upon receiving the *OT factors*, the license server creates *n* licenses for each item in *I* with the user's id and the corresponding rights. These licenses are encrypted with the user's public key or the secret key shared with the user's smart card. Then server will reply the *OT result* with $(y,\{D_i \mid 1\le i\le k\},\{c_j \mid 1\le j\le n\})$. The user can retrieve the encrypted licenses from the *OT result*. These encrypted licenses will be passed to a smart card or trusted software for decryption. After obtaining the licenses, the user may render digital content on his wishes.

If the license server receives a *License Update* request, it will refresh the right of each license in *I* and encrypt those licenses using the same key. Then it will retrieve the *OT factors* from the database and calculate the new *OT result* to reply the user. The user cannot obtain other licenses except what he has bought since the *OT factors* are recorded in the server database. The user will have the same choices if the *OT factors* remain unchanged.

In addition, the user should store the (σ_i, a_i) pairs for license updating after sending the license request message. The pairs can be stored in the user's smart card, if the purchased contents are not too many. However, the increasing numbers of (σ_i, a_i) pairs may cause the exhausted storage of the smart card. The alternative method is to encrypt the pairs with the user's public key and send to the license server together with the license request message. When the *License Update* protocol is activated, the pairs will be sent from the license server to the user together with the new *OT result*. After all, the user can compute the new license successfully.

In this basic scheme, the license server cannot know the choices of the user (what it know is the user pick *k* out of *n*). We define *E* be the event that license server correctly guesses what the user picks. Here we give a measurement of privacy call *Privacity* as defined as:

$$\text{Privacity P} = 1-\Pr(E) = (n-k)/n$$

If *k* is small, we have a large *Privacity* which implies that the user has a high privacy, and vice versa. We can immediately follow that a larger *n* always provides a better privacy. However, a larger *n* will also cause a higher computation and transmitting cost to the server and user. Therefore, we should design the system in the fashion that achieves an acceptable *Privacity*, (e.g. 0.9) for a fixed *k*, (e.g. 3) and minimizes *n*.

Now, suppose the license server has 30*n* contents for sales. These contents may have the same price but different types, varies from classic music to children movie. The license server should group *n* contents as a group and together have 30 groups of contents. When the user purchases licenses from the license server, he should also send the *group id* of the contents they want to buy together with the *OT factors*.

We remark that each group should consist of different types of contents to preserve users' privacy. Usually the type of the contents is more sensitive in the user's privacy. Also, dividing contents into groups allows contents to have different prices. For each group having the same prices, license server will be able to sell contents with different values and not able to know what the user bought.

V. Security Analysis

In this section, we will evaluate how secure our protocol achieves. We summarize the evaluation here – our protocol is resistant against: license cracking, license stolen, and privacy exploration against malicious server. We will discuss their importance and how our protocol achieves each of those.

A. License Cracking

Attacker may try to steal the content key from a license to decrypt the content, or modify a license to have a different right (e.g. extending the expiration date). Not only because licenses are encrypted, the OT scheme has also masked the encrypted licenses in the *License Acquire*-

ment phase and the *License Update* phase; therefore, passive adversary cannot extract the session key from the license. Moreover, since licenses are signed, rights cannot be forged by attacker; otherwise, the player will not render the media content correctly.

B. License Stolen

A malicious user will try various ways to obtain license that he has not purchased. The OT scheme guarantees malicious users are not able to obtain more than *k* licenses for a particular choice of OT factors. In the license update phase, the same OT factors are employed, therefore, the choices of contents are "fixed" in the *License Acquirement* phase. As discuss in part a), users are not allowed to forge licenses.

C. Privacy against malicious server

If contents are sensitive to an autocratic government, the license server may be forced to figure out who purchase this content. However, the OT scheme preserves users' privacy with a certain value of *Privacity*. Even if the government suspects someone who accesses this content, there will have no significant evident to prove this person access this content. However, we may allow the license server to know how many people aware of this content. We will discuss that in the section VII.

VI. Comparisons

We compare our application with three previous schemes about privacy protection in this section. These privacy protection schemes respectively are Park *et al.*'s privacy enhancing protocol (*PrecePt*) [24], Grimm *et al.*'s approach [11], Lee *et al.*'s scheme [15], and Conrado *et al.*'s method [5]. The comparisons among these schemes are shown in Table 1.

Table 1: Comparison among other privacy preserving DRM systems

	[24]	[11]	[15]	[5]	Our
Privacy	×	×	×	×	○
Anonymity	○	○	○	○	×
Applied Scheme	Temporal User ID	Pseudonymous ID	E-cash system	Hashed public key	Oblivious Transfer
License Form	Separate	Separate	Separate	Separate	Separate
License Update	×	×	×	×	○
Payment System	e-cash	e-cash	e-cash	e-cash	various choices
Membership Discount or Club Coupon	×	×	×	×	○
Accounting	×	×	×	×	○

The main difference between our application and the other schemes is that we provide privacy but the others provide anonymity. Each of the schemes uses various methods to realize anonymity and privacy. For instance, Park *et al.* and Grimm *et al.* use a temporal user id (or called a pseudonymous consumer id) to instead of the user id. Next, Lee *et al.* apply the e-cash system to DRM system for anonymity. Third, Conrado *et al.* hash the user's public key to keep the original one secret. Finally, our application applies the OT scheme to reach the privacy.

Moreover, the licenses in these schemes are all distributed separately with the contents. However, only our application provides *License Update* protocol for the license server to maintain the usage rights correctly. Furthermore, the payment system is various in our application but the others only can apply e-cash payment system. License server is capable to manage its users in a more effective fashion. It not only can provide discount to frequent buyers, but also can trace and prohibit malicious customers. Also, anonymous trade will leak users' related information via side channel like IP addresses. In conclusion, our proposed application is more profitable and powerful than the previous schemes.

VII. Practical Issue

In this section, we discuss the practical issue about the proposed application. In practice, many details of this application must be considered carefully to make sure the application will work well. For instance, the commercial issue about the different prices of the contents or the practical issue about computational power of the device will be mentioned below.

A. On-Line/Off-Line DRM System

Usually we call a DRM system is on-line if the device needs to connect to the server for updating the license when a user is going to access the content. On the contrary, if the device does not need to connect to the server when the content is accessed, the DRM system is called off-line. This is the difference between the on-line DRM system and the off-line DRM system.

The user in the on-line DRM system should purchase only a single content in each execution of the *License Acquirement* protocol. Otherwise, the license will be incorrectly updated. If the user in the on-line DRM system purchases two or more contents, the server will update all licenses of the same group and then create a new *OT result* during the *License Update* protocol. The user, consequently, will receive this new *OT result* and compute it to get the corresponding licenses, but the rights of each license is modified. Although you only want to update one license, now the others in the same transactions are updated, too. Therefore, the user in the on-line DRM system should only request a single license in each transaction. On the other hand, the user in the off-line DRM system is allowed to purchase unlimited contents in each transaction because the license needs not to be updated.

B. Different Prices

The group of the license is classified by the price, so that the contents in the same group are marked identical prices. In addition, the privacy in our application relies on the amount of licenses in the classified group. If the *Privacity* is small, the server can guess easily which contents the user may buy. However, in the real world the different contents usually are marked different prices. That causes a small amount of licenses in the same group, and then violates indirectly the users' privacy.

In order to solve this problem, we might set the price of a content be a multiple of unit price. For example, if the prices of a list of contents are $5, $10, $15, and $30, we set the unit price be $5 and make the prices of the contents be 1, 2, 3, and 6. Then, we split the licenses into the number of unit prices by the technique secret sharing [30]. Therefore each part of the license has a corresponding *License ID*. For instance, a user downloads a content which costs $25, and he will need to purchase the content with 5 *License ID* to help him to purchase five license fragments. After receiving the 5 pieces of license fragments, he can compute the original license.

C. Payment System

In the previous privacy protection schemes, they always conceal the users' information from the license server. Due to this anonymity, the only way to charge in their protocol must use the electronic cash payment system. It is inconvenient for users to get the e-cash before the transactions. In addition, the e-cash also has risky to be stolen.

However, the users in our application can pay easily and safely. When the users request the license, they also provide their *User ID*s. Because the users have registered before, their accounts can be recorded in the license server. Therefore, only the users should do is waiting the bill to pay. Of course, the users in our application also can still choose the e-cash system as their payment system.

D. Computation Load Balancing

Due to OT scheme, the computation of our application is very large. This seems not suitable for the low-computational-power device, such as mobile phone [17]. However, the *OT factors* can be computed independently because this computation only preserves privacy from the license server. Therefore, the computer can help to compute this computation. For instance, when the low-computational-power device needs to compute the corresponding *OT factors*, it can connect with a PC. Then, the *OT factors* will be calculated by the PC and be transmitted to the low-computational-power device. Although the PC knows the *OT factors*, the licenses are encrypted with the users' key and thus the PC cannot obtain the licenses. The PC is only trusted to not to release the secret value of *OT factors* to the others to expose the privacy of the user.

E. Size of a group

To maintain a certain level of *Privacity*, the size of group *n* and the number of choice *k* have to be carefully selected. The following figure shows the variations of *Privacity* against different choices of *n* and *k*.

In order to preserve certain level of privacy for users, we recommend users to choice a *Privacity* be at least 0.95. We may learn from figure 4, for instance, if all the contents in the group are the same price, we may set the group size be 20. If the prices of the contents in the group are different, such that *k* may varies from 1 to 5, we should set *n* be at least 100. We remark that a larger *n* will cost a higher computation in server side and higher transmitting cost.

F. Products Statistics

In our application, the server only knows which groups of contents a user purchases. However, it can be possible to gather the information of the purchased contents. This information will be helpful for the servers to know what content is the best sale. We can, of course, obtain this information from the content server. But if the content server is uncooperative or lacking management (like peer-to-peer system), we can also do some tricks on the license server to obtain the sales statistics. Every user has a different combination of the licenses in the license group. When a user has purchase a license from a license group, all the counters of each license of the group will be increased by one. The server still does not know what contents the user buys actually, but it can be possible to conclude the best sale of the contents, as long as the combination of the licenses for each user is ingenious.

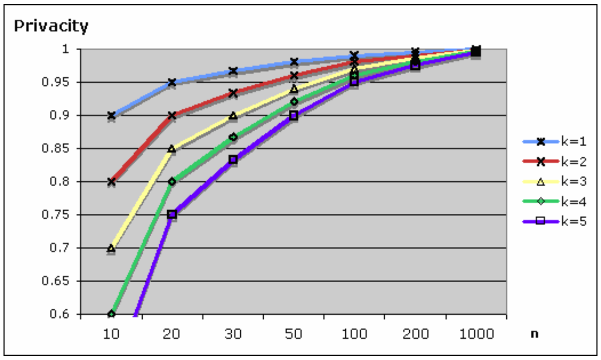

Fig 4. Privacity Graph

VIII. Conclusions

In this paper, we present a new scheme for handling privacy in DRM systems. It makes sure that servers cannot know what contents a user buys but know the user's ID or other related information. Many papers discuss the privacy issue; however, all of them only provide "anonymity". In their schemes, the servers do not recognize the users. Although both privacy and anonymity protection can prevent the server to know the relations between the users and the contents, it is more profitable by using privacy instead of anonymity.

Our application supports various payment systems and makes the transaction more easily and safely. Although the server does not know the information about the contents that the users buy, the products statistics can still be gathered. In addition, our application can be applied to most of the existing DRM systems. We believe that our privacy protection application will be applied more popular in the near future.

References

[1] Apple iTunes: http://www.apple.com/itunes/

[2] M. Bellare and S. Micali, "Non-interactive oblivious transfer and applications," in *Proceedings of Advances in Cryptology* – CRYPTO'89, Vol. 435, pp. 547–557, 1989.

[3] J.E. Cohen, "DRM and Privacy," *Communications of the ACM*, Vol. 46, No. 4, pp. 46-49, April 2003.

[4] C. Conrado, F. Kamperman, C.J. Schrijen, and W. Jonker, "Privacy in an Identity-based DRM System," in *IEEE Proceedings of the 14th International Workshop on Database and Expert Systems Applications (DEXA'03)*, Prague, pp. 389-395, September 2003.

[5] C. Conrado, M. Petkovic, W. Jonker, "Privacy-Preserving Digital Rights Management," *Secure Data Management*, pp. 83-99, 2004.

[6] Council of Europe Convention 108: http://conventions.coe.int/treaty/EN/Treaties/Html/108.htm

[7] C.K. Chu and W.C. Tzeng, "Efficient k-out-of-n oblivious transfer schemes with adaptive and non-adaptive queries," in *Proceedings of International Workshop on Practice and Theory in Public-Key Cryptography (PKC'05)*, Lecture Notes in Computer Science 3386, pp. 172-183, 2005.

[8] Electronic Media Management System (EMMS), Available: http://www.ibm.com/software/emms

[9] Electronic privacy information center (2004, March 29), "*Digital Rights Management and Privacy*," Available: http://www.epic.org/privacy/drm/

[10] J. Feigenbaum, M. Freedman, T. Sander, A. Shostack, "Privacy Engineering for Digital Rights Management Systems," *DRM 2001*, Lecture Notes in Computer Science, Vol. 2320, Springer, Berlin, 2002, pp. 76-105.

[11] R. Grimm, P. Aichroth, "Privacy protection for signed media files: a separation-of-duty approach to the lightweight DRM (LWDRM) system," in *Proceedings of the 2004 multimedia and security workshop on Multimedia and security* (MM&Sec'04), Magdeburg, Germany, pp. 93-99, September 2004.

[12] R. Iannella. ,"Digital Rights Management (DRM) *Architectures," D-Lab Magazine, Vol.7 No.6*, June 2001

[13] InterTrust, Available: http://www.intertrust.com/

[14] S. Kenny and L. Korba, "Applying digital rights management systems to privacy rights management", *Computers & Security*, Volume 21, Number 7, November 2002, pp. 648-664

[15] D.G. Lee, H.G. Oh, and I.Y. Lee, "A Study on Contents Distribution Using Electronic Cash System," in *Proceedings of the 2004 IEEE international Conference on e-Technology, e-Commerce and e-Service* (EEE'04), pp. 333-340, March 2004.

[16] M. Lesk, "The good, the bad, and the ugly: what might change if we had good DRM," *Security & Privacy Magazine, IEEE, Vol. 1, Issue:3*, pp. 63-66, May 2003.

[17] T.S. Messerges, E.A. Dabbish, "Digital Rights Management in a 3G Mobile Phone and Beyond," *ACM DRM'03*, Washington, DC, USA, pp. 27-38, October 27, 2003.

[18] Microsoft DRM: http://www.microsoft.com/ windows/windowsmedia/drm/default.aspx

[19] Y. Mu, J. Zhang, and V. Varadharajan, "m out of n oblivious transfer," in *Proceedings of the 7th Australasian Conference on Information Security and Privacy (ACISP'02)*, Vol. 2384, pp. 395–405, 2002.

[20] D. Mulligan, J. Han, and A. Burstein, "How DRM based content delivery systems disrupt expectations of "personal use"," *In Proceedings of the 2003 ACM workshop on Digital Rights Management (2003)*, ACM, pp. 77–89.

[21] M. Naor and B. Pinkas, "Oblivious transfer with adaptive queries," in *Proceedings of Advances in Cryptology* – CRYPTO'99, Vol. 1666, pp. 573–590, 1999.

[22] Official Journal L 281, 23/11/1995 p. 0031 – 0050.

[23] W. Ogata and K. Kurosawa, "Oblivious keyword search," *Journal of Complexity*, Vol. 20, pp. 356–371, 2004.

[24] B.N. Park, J.W. Kim, and W. Lee, "PrecePt: a privacy-enhancing license management protocol for digital rights management," in *Proceedings of the 18th International Conference on Advanced Information Networking and Application* (AINA'04), pp. 574-579, August 2004.

[25] M.O. Rabin, "How to exchange secrets by oblivious transfer," *Technical Report TR-81*, Aiken Computation Laboratory, Harvard University, 1981.

[26] H.M. Sun, C.F. Hung, and B.H. Ku, "An Improved Identity-Based DRM System," *Proceeding of Information Security Conference (ISC) 2005*.

[27] M. Ter Maat, "The economics of e-cash," *Spectrum, IEEE, Vol.34, Issue:2*, pp. 68-73, February, 1997.

[28] W.-G. Tzeng. "*Efficient 1-out-of-n oblivious transfer schemes with universally reusable parameters*," IEEE Transactions on Computers 53(2), pp.232-240, 2004.

[29] W. W. Ware, "Records, Computers, and the rights of citizens," *Advisory Committee on Automated Personal Data Systems*, July 1973

[30] S.Y. Yan, *Number Theory for Computing, 2nd Edition*, Berlin: Springer, 2002, pp. 399-403.