23

Enhancing Privacy for Digital Rights Management

Milan Petković, Claudine Conrado, Geert-Jan Schrijen and Willem Jonker

Philips Research
The Netherlands

Summary. This chapter addresses privacy issues in DRM systems. These systems provide a means of protecting digital content, but may violate the privacy of users in that the content they purchase and their actions in the system can be linked to specific users. The chapter proposes a privacy-preserving DRM system in which users interact with the system in a pseudonymous way, while preserving all the security requirements of usual DRM systems. To achieve this goal, a set of protocols and methods is proposed for managing user identities and interactions with the basic system during the acquisition and consumption of digital content. Privacy-enhancing extensions are also proposed. Unlinkable purchase of content, which prevents content providers from linking all content purchased by a given user, is discussed. Moreover, a method that allows a user to transfer content rights to another user without the two users being linked by the content provider is provided.

23.1 Introduction

Thanks to the Internet, which provides an excellent trading infrastructure, nowadays digital content distribution has become one of the most quickly emerging activities. As a consequence of this trend and the success of one of the first online music shops, Apple's iTunes, which has recently sold its 500 millionth song [1], a number of shops have been opened [2-6] and both consumers and content providers have clearly shown great interest in electronic distribution of audio and video content.

Digital content can, however, be easily copied, exchanged and distributed illegally, which is obviously a threat for the content industry. This has triggered active research on technologies that can protect digital content from illegal use. One of the most important of these technologies is digital rights management (DRM) technology that provides content protection by enforcing the use of digital content according to granted

rights. It enables content providers to protect their copyrights and maintain control over the distribution of and access to content. The most widely used DRM systems nowadays in the mainstream entertainment arena are Microsoft Windows Media DRM 10 [7] and Apple's FairPlay [8], which are the two big players for PC-centric music services. Other DRM systems are Sony's Open MagicGate [9], Helix from RealNetworks [10] and Thomson's SmartRight [11].

Early DRM systems were device based, which means that rights were bound to devices and content was only accessible on a specific device. However, in order to allow a consumer to access his content anytime, anywhere, on any device, the idea of person-based DRM has emerged, as discussed in Chap. 20. Furthermore, some companies are investigating new concepts such as authorized domains [12-14] and personal entertainment domains (PEDs) [15], which take into account (along with the requirements of content owners) the requirements of content consumers. In PEDs, for instance, content can freely flow inside a domain (typically a household), so that it can be freely copied inside that domain and exchanged among the domain devices. However, the system controls transactions between different domains.

To protect the content and enforce the rights given in a license, a DRM system normally identifies a user and monitors the usage of content. Therefore, DRM systems are very privacy-invasive, violating the users' privacy in many ways. For example, they do not support anonymous and un-linkable buying or transfer of content as in the traditional (physical) business model where a user anonymously buys a CD using cash. Furthermore, they generally involve tracking of the usage of content in order to enforce the rights [16,17]. In person-based DRM systems, e.g., a user has to authenticate himself each time he accesses a piece of content. Therefore, information such as user identification, content identification, time and place of access, etc., can be collected. The same holds for device-based DRM systems, except that user identification may not be straightforward, but through other data that can be linked to the user.

As privacy is becoming increasingly important in the connected digital world, the possibility of creating user profiles or tracking users creates numerous privacy concerns. In order to overcome the aforementioned privacy problems in DRM systems, this paper proposes several methods to enhance privacy. The main idea is to allow a user to interact with the system in an pseudonymous way during the whole process of buying and consuming digital content. This has to be done in a way that all the security requirements of the usual DRM systems are satisfied. This means that content providers must be assured that content is used according to issued licenses and cannot be illegally copied. Furthermore, we discuss a solution that prevents the linkability of purchase actions by anonymous users. Finally, an approach is presented to anonymously transfer licenses, so that a piece of content can be sold or gifted to another user without the content provider being able to link the two users.

The remainder of the chapter is organized as follows. In Sect. 23.2, the basic privacy-preserving DRM (PPDRM) system is introduced. Section 23.3 discusses a solution that extends the basic system to support unlinkable purchase of content. In Sect. 23.4, the system is extended to support anonymous transfer of licenses. Finally, Sect. 23.5 draws conclusions.

23.2 Basic System

In the basic PPDRM system, a user is represented by means of pseudonyms, which are decoupled from the user's real identity. Based on these pseudonyms, the system tackles a number of threats to the privacy of the users of this system, and also related threats to the security of the system. These threats are mentioned below and are handled by the PPDRM system by means of protocols discussed in the next sections.

The association between a user's real identity and content owned by the user is the main privacy threat circumvented by PPDRM. This association may happen if personal licenses are used for content access, and it allows the tracking of users while they access content. To avoid that, the system exploits persistent (i.e., long-term) user pseudonyms.

A common security threat in DRM systems is the hacking of devices, e.g., personal smart cards and devices on which content is accessed. The PPDRM system avoids this threat by means of compulsory mutual compliance checks between smart cards and devices. Such checks, however, may violate users' privacy. To avoid that, the system exploits temporary (i.e., short-term) user pseudonyms.

Although users do not disclose their real identity in the system, there is still a threat to their privacy, which is the linkability of a user's content purchase actions via his persistent pseudonym. The PPDRM system deals with this problem by means of a mechanism which allows users to renew their persistent pseudonyms. The system also prevents the user from misusing the system by transferring their licenses to others.

Finally, the transfer of licenses between users causes important security and privacy threats. For example, a user may be able to continue using his licenses after he has transferred them to another user. Concerning privacy threats, the association between the user who transfers and the user who receives a given license is typically disclosed. To avoid these threats, the PPDRM system makes use of invalidation lists and anonymous licenses issued by the CP.

Entities in the basic PPDRM system include the *user*, the *content provider* (CP) and the *compliant device* (CoD), a device that behaves according to the DRM rules. Related to the CoD, there is the *compliance certificate issuer for compliant devices* (CA-CoD). Moreover, there is the *smart card* (SC), which is the user ID device. In the following sections,

where no confusion may be caused (e.g., in the description of protocols), the user and his smart card are referred to interchangeably. Related to the smart card there are the *smart card issuer* (SCI) and the *compliance certificate issuer for smart cards* (CA-SC).

Figure 23.1 depicts the different transactions performed involving the entities mentioned above. These transactions and different aspects of the system are described in the sections below, where references to the numbered links in Fig. 23.1 are made at the appropriate points.

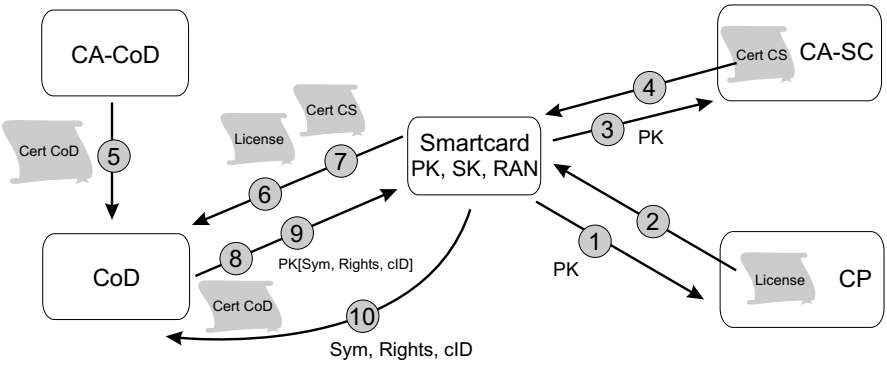

Fig. 23.1. Interactions among different entities of the PPDRM system.

23.2.1 Acquisition of a Smart Card by the User

The acquisition of a smart card by the user is done in an anonymous way. The user buys a smart card from a retailer, taken from a pool of identically looking smart cards pre-issued by the SCI. Each smart card contains a different public-private key pair (PK, SK) and an unset personal identification number (PIN), e.g., all PINs are set initially to 0000. The SCI guarantees that, as long as the PIN is unset, the public key of that specific card is not revealed to any party. So, when the user interacts for the first time with the card, he is asked to set a PIN, after which the card becomes active and reveals its public key PK. The PIN can never be reset back to the null value, so the user is sure that he is the first one to learn that PK. Once set, the PIN can be used to activate the card to allow its engagement in transactions with other entities. The PIN should be kept secret by the user, which guarantees that the card can only be used by *that* user. This activation procedure is assumed and will not be explicitly mentioned in the smart card's transactions in the remainder of this chapter.

With the setup above, no one should be able to make an association between the user's real identity and the PK. Note that the private key SK is securely stored on the smart card and is not accessible to the user nor to any other party (except of course the SCI). This is a crucial security aspect

of the system. As can be seen in the next section, the leakage of the SK would allow the user to, e.g., freely distribute all (unencrypted) content for which he bought a license.

Security assumptions in this context are (i) the public key PK of a SC is revealed and the PIN number is set only after the first transaction, and (ii) the private key SK corresponding to the public key PK is stored secretly and only known to the SC.

23.2.2 Acquisition of the Content and the Rights by the User

The acquisition of content and licenses is performed as follows. The user's SC contacts the CP with the request via an Internet connection using an anonymous channel. This can be implemented, e.g., via a mix network [18] or a simpler proxy service [19]. The anonymous channel hides the user's IP address and prevents the user identity from being derived from the IP address. After an anonymous payment scheme is conducted (such as the pre-payment scheme described in [20]), the user's SC sends the public key PK to the CP (link 1 in Fig. 23.1). It is assumed that the SCI keeps track of all smart cards it has issued and of their behavior by means of a revocation list with the PKs of hacked SCs. With this setting, the CP can check with the SCI whether PK is legitimate and whether it belongs to the revocation list or not. If it does not, the CP can create the right or license for that content. The content itself is encrypted by the CP with a symmetric key, Sym, randomly chosen by the CP, and sent to the user (link 2 in Fig. 23.1) together with the license, whose format is given in (1). Both, content and license, can then be stored by the user, e.g., on an optical disk or personal device.

$$\{\, PK[Sym//Rights//contentID] \,,\, H(Rights//contentID) \,\}_{signCP} \qquad (1)$$

In the license above, PK encrypts the concatenated value [Sym//Rights//contentID], where Rights describe the rights bought by the user, contentID identifies the content and signCP is the CP's signature on the certificate. The hash of (Rights//contentID) is also added to the license to allow a compliant device to check these values upon a content access request (as discussed in Sect. 2.4). The CP's signature on both terms in the license guarantees that these terms have indeed been created by the CP. Moreover, given that the PK encrypts the value [Sym//Rights//contentID], the SC is the only entity that can obtain the key Sym from the license by using the private key SK. Furthermore, a compliant SC (as attested by the compliance certificate discussed in the next section) will reveal the key Sym only to a compliant device during the content access action.

The license in (1), if seen by any party, e.g., on the user's optical disk, does not reveal the public key PK nor the rights, nor the content identifier,

so it preserves the user's privacy with respect to content and rights ownership. Therefore, if found in the user's possession, it does not compromise the user's privacy. Note, however, that an eavesdropper may have been able to associate the public key PK sent to the CP with the license sent back by the CP during a buying transaction, if these values were sent on the clear. Therefore, the anonymous channel used should also be secret, i.e., the exchanged data should be encrypted.

The CP learns the association (PK↔(contentID, Rights, Sym)) during purchase, but not the user's real identity due to the anonymous channel.

Security assumptions in this context are (i) there is a mechanism in place to allow the user to pay anonymously for the license he requests, (ii) the user contacts the CP via an anonymous channel, (iii) the channel is also secret, and (iv) the SCI is responsible for keeping track of hacked SCs (i.e., those whose secret key SK has been revealed or whose functionality has been changed in any way).

23.2.3 Acquisition of SC Compliance Certificate by the User

To ensure security of the protocol between the user's SC and the CoD, a mutual compliance check is performed. That means that the SC checks the CoD's compliance but must also show an SC's compliance certificate to the CoD. The acquisition of this certificate is described below.

The SC's compliance certificate does not contain the user public key PK, but is issued by the CA-SC with a frequently renewed SC's pseudonym, for reasons given below. To obtain this certificate, the user's SC contacts the CA-SC via an Internet connection using an anonymous channel, as in the interaction with the CP above. Again, the anonymous channel used must be secret to prevent eavesdropping on the channel. The user's SC sends its public key PK (link 3 in Fig. 23.1) with a request for the certificate, and the CA-SC checks with the SCI whether PK belongs to the revocation list or not. If it does not, the CA-SC generates a pseudonym for the SC, say a random number RAN, and issues the following compliance certificate, which is sent to the SC (link 4 in Fig. 23.1):

$$\{H(RAN)\,,\, PK[RAN]\}_{signCA\text{-}SC}\,, \tag{2}$$

where H is a one-way hash function, PK encrypts RAN and signCA-SC is the signature of the CA-SC on the certificate.

The compliance certificate above does not reveal the public key PK nor the SC's pseudonym RAN. Furthermore, the only entity which can obtain RAN from the certificate is the SC (by decryption with the private key SK). The value RAN may then be checked by the device via the hash value in the certificate. The use of a pseudonym RAN allows the device (verifier) to check the compliance of the SC without learning its public key PK from the certificate. Moreover, the linkability of different shows of a

given SC's compliance certificate can be minimized since a frequent renewal of the compliance certificates (and, as a consequence, of the pseudonyms RAN) is a requirement of the DRM system. This can be achieved by including an expiration date in the compliance certificate.

On the other hand, there are different methods to prevent the linkability of pseudonyms. For example, the convertible credentials described in [21] allow a user to obtain a credential from a given organization under a given pseudonym, and show that credential to another organization under another pseudonym. This type of approach involves protocols which are significantly more complex than the simple protocols described in this paper, which involve only simple hash operations.

During the acquisition process of the compliance certificate, the CA-SC learns the association (PK↔RAN), but not the user's real identity due to the anonymous channel.

Security assumptions in this context are (i) the user contacts the CA-SC via an anonymous channel, (ii) the channel is also secret, and (iii) the SCI is responsible for keeping track of hacked SCs.

23.2.4 Access to Content by the User

Finally, the user can access the content for which he bought the license. This can be performed on any CoD, which may be trusted or untrusted by the user (note that the discussion below on possible compromises to the user's privacy is only relevant in the case of *untrusted* CoDs). The encrypted content and the license may both be stored, e.g., on a user's portable device. Alternatively, the license (but likely not the encrypted content) may be stored in the user's SC. Whichever the case, license and content are both transferred to the CoD (link 6 in Figure 23.1). This allows this device to check the license (as described below), and further decrypt and render the content to the user if allowed. But before that happens, a mutual compliance check must be performed as described next.

The CoD proves its compliance by means of a CoD compliance certificate. This certificate is issued by the CA-CoD (which certifies the CoD's public key) and sent to the CoD beforehand (link 5 in Figure 23.1). Upon the compliance check, the certificate is shown to the SC (link 8 in Fig. 23.1). The SC must therefore store the public key of the CA-CoD. This key may be changed periodically, which obliges the CoD to periodically renew its compliance certificate, thus allowing revocation of CoDs. This solution is preferred to that of including an expiration date in the CoD compliance certificate, as the SC may not have a clock. Moreover, periodic change of the CA-CoD's public key also implies that the SC must renew that key periodically. This could be done, e.g., when the SC obtains its own compliance certificates from the CA-SC, as this authority could also safely provide the SC with the CA-CoD's public key. Once the CoD has been checked, the SC proves its compliance by showing the

pseudonymous compliance certificate in (2) to the CoD (link 7 in Fig. 23.1). As mentioned above, the SC can obtain the value RAN and send it to the CoD which checks the value via the term H(RAN). Since the CoD can have a clock, the SC compliance certificate may contain a time of issuance and a validity period added to it, which obliges the SC to periodically renew the certificate when it gets too old. Note that it is also in the interest of the SC to renew its compliance certificate often enough so as to minimize the linkability mentioned above.

If the mutual compliance check is positive, the CoD sends the term PK[Sym//Rights//contentID] from the license to the SC (link 9 in Fig. 23.1), which then decrypts the term and sends the values Sym, Rights and contentID back to the CoD (link 10 in Fig. 23.1). Note that, although the compliance of the SC is checked by the CoD, it is always possible that a dishonest SC has not yet been detected. Therefore, to ensure that the SC sends the correct values of Rights (and contentID), the CoD checks the hash value in the license which has been previously transferred to it. Only if it is correct, the CoD uses Sym to decrypt the content and gives the user access to it, according to Rights.

During access to content by the user, the CoD learns the association (RAN↔(contentID, Rights, Sym)). The CoD may also learn the user's real identity, as the user is now physically present in front of the CoD (e.g., the CoD may have a camera). However, the public key PK of the user is never revealed to the CoD at the time of content access. Therefore, this compromises the user's privacy only concerning the specific content and rights involved in the access transaction. The threat is of course higher if the user accesses many different pieces of content on the same CoD. This type of attack cannot really be avoided. Considering this is not the case, what the proposed mechanism prevents is that a content access action by a user on a CoD, possibly under the control of an attacker, may easily allow the attacker to learn all other content bought by the user. Moreover, if the attacker does not learn the user's real identity, the mechanism limits the number of transactions for which the user may be tracked by a given CoD, as RAN changes often.

Security assumptions in this context are (i) the CA-CoD is responsible for keeping track of the CoD's behaviour as well as for issuing compliance certificates for those devices, (ii) a compliant SC will send the right values and only reveal the decryption key Sym to a compliant device (CoD), and (iii) the CoD will not reveal the key Sym to any party, except for perhaps another (proven) compliant device.

23.3 Non-linkable Purchase of Content

In this section, the basic PPDRM is extended to prevent linkability by the CP of content purchased by a given user with public key PK. Linkability

may compromise a user's privacy if the association between PK and his real identity is disclosed to the CP for at least one piece of content. This means that the association is disclosed for all content bought by that user. The solution is based on user pseudonyms, which can be used to buy different pieces of content, and includes the steps of pseudonym certification and anonymous purchase.

23.3.1 System Assumptions

It is assumed that users have a Diffie-Hellman key pair and that from the original public key new public keys are derived, which can be certified by a trusted certification authority (referred to as CA). The system parameters g, p and q are chosen as in general Diffie-Hellman key agreement [22], with g referred to as the group generator. The user's private key is $SK \in [1,q\text{-}1]$ and the corresponding public key is generated as $PK = g^{SK} \bmod p$. For brevity, the modulo operation will be omitted in the remainder of this section.

With the assumptions above, public key encryptions can be implemented as El-Gamal encryptions [23]. For signing messages, the digital signature standard (DSS) [24] with the digital signature algorithm (DSA) can be used since it uses Diffie-Hellman keys. The reader may also refer to [25] for more details on the cryptographic tools and protocols mentioned in this section.

23.3.2 Pseudonym Acquisition and Certification

The user must have his pseudonyms (in the form of new public keys) certified at the CA before he can use them to buy content rights from the CP. The communication steps between the user and the CA are explained below and depicted in Fig. 23.2.

The user sends his original public key PK to the CA, which allows the CA to check with the SCI whether PK is legitimate and whether it belongs to the revocation list or not. If all checks are successful, the two parties proceed to establish a secure authenticated channel (SAC). Next, the user creates a random value α and sends it securely to the CA (alternatively, the CA may generate α and send it to the user). With α and PK, the CA creates the pseudonym PK^* by raising PK to the power α, i.e., $PK^* = PK^{\alpha} = g^{\alpha SK}$. The new public key PK^* is created in this way for reasons discussed below. Next, the CA creates and signs a digital certificate containing the pseudonym PK^* and securely sends it back to the user's SC. This certificate proves that the pseudonym PK^* belongs to a user with a legitimate PK. It is assumed that the CA keeps track of all pseudonyms generated from (i.e., associated with) a given public key, but that it keeps this information confidential. This is only disclosed to the SCI if this

authority discovers that the SC with PK^* has been hacked. This allows the SCI to add that SC to the revocation list, by entering not the pseudonym PK^* but its original public key PK to the list.

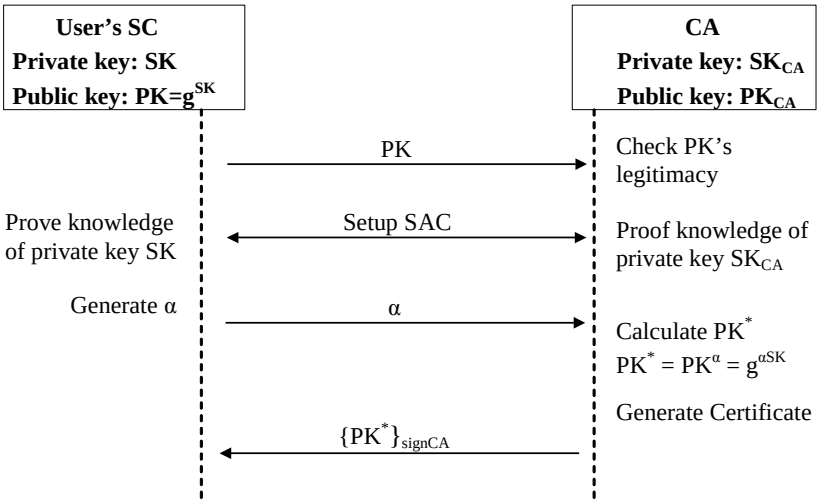

Fig. 23.2. Protocol for acquisition and certification of a pseudonym (new public key) by a user with the CA.

The new public key PK^* corresponds to a new private key SK^* which can be easily computed by the SC as $SK^* = \alpha SK$. Moreover, only the SC (and no other party, including the CA) can calculate this key, so SK^* can be kept secret by the SC in the same way as the original private key SK. With this crucial security aspect of the system taken into account for pseudonyms as well, the CP can issue pseudonymous licenses for content access with the format shown in (1), as it will be seen in the next section. Therefore, once calculated by the SC, the new key SK^* must be securely stored (i.e., no party should be able to access that key in the SC).

23.3.3 Content Rights Purchase

The purchasing procedure is similar to the procedure described in Sect. 2.2. It is explained below and illustrated in Fig. 23.3.

After the user contacts the CP via an anonymous channel requesting the rights to given content under a given pseudonym, an anonymous payment scheme is conducted. The pseudonym certificate is then sent to the CP, which checks the signature on the certificate. If it is correct, the CP can issue a license as shown in (3) with the pseudonym PK^* as subject, which can then be sent to the user. Note that this license has the same format as the license given in (1).

$$\{ PK^*[Sym//Rights//contentID] , H(Rights//contentID) \}_{signCP} \quad (3)$$

As noted before, to prevent an eavesdropper from being able to associate the public key PK^* sent to the CP with the license sent back by the CP during the buying transaction, the communication channel must be secret.

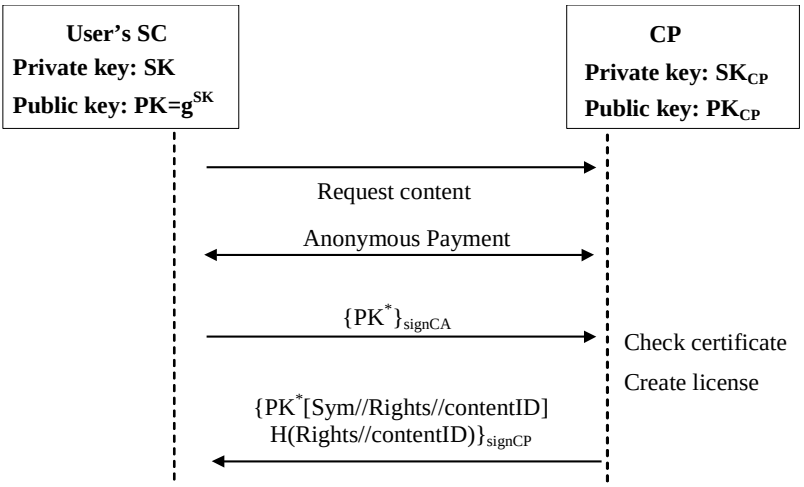

Fig. 23.3. License purchase by a user under a pseudonym PK^* certified by the CA.

23.3.4 Content Access

Once in possession of the license as given in (3), the user can access the content on any CoD. The encrypted content and the license are transferred to the CoD, which then performs a mutual compliance check with the SC. The CoD compliance certificate and the SC pseudonymous compliance certificate are described in Sect. 2.4. The latter is issued with a dynamic value RAN and is obtained from the CA-SC under public key PK (i.e., PK, and not PK^*, encrypts the value RAN in the certificate).

As before, after the mutual compliance check, the CoD sends PK^*[Sym//Rights//contentID] to the SC, which decrypts it using SK^*. The values Sym, Rights and contentID are then sent back to the CoD. If the value H(Rights//contentID) checks with the received values, the CoD decrypts the content and gives the user access to it in accordance to Rights.

23.4 Anonymous Transfer of Licenses

A user should be able to transfer his license to another user. This transfer must be done in a way that prevents the original owner from still being able to access the content after the transfer by using the license. It is further required here that the transfer be anonymous, i.e., no party learns the association between the two users. Therefore, a solution is discussed below which extends the basic PPDRM to tackle license invalidation and license anonymization. The same procedure applies if the user bought his license under a pseudonym, which simply replaces PK in all interactions with the CP.

23.4.1 License Invalidation

To allow a user (referred to as the first user) to transfer his license, he contacts the CP via an anonymous channel, authenticates with his public key PK, presents the license to be transferred to the other user (referred to as the second user) and provides the second user's public key PK'. Note that here the CP learns the connection between the two users. The CP marks that license with PK as "to be invalidated", so before the CP creates a new license with PK', invalidation of the old license must be dealt with.

The invalidation problem can be solved by including in the compliance certificate of the first user's SC a list with all the licenses that are to be invalidated. This can be done when the SC obtains its compliance certificate. The CA-SC contacts the CP and asks for that list for PK. The CP uses the symmetric key Sym_i to identify a given invalidated license i, and creates a list with the values H(Sym_i // Time). H( ) is a one-way hash function used to conceal the values of Sym_i and to reduce the size of the terms in the invalidation list, and the current time (Time) is concatenated with each Sym_i to prevent the linkability of compliance certificates issued for PK in different occasions. Once the list with H(Sym_i // Time) values and the value Time are sent to the CA-SC, the CP considers as resolved the invalidation of the licenses of PK and can create the new license for the second user, which includes public key PK'. The SC's compliance certificate now has the format as in (4).

$$\{ \text{H(RAN)}, \text{PK[RAN]}, \text{Time}, \text{H(Sym\_1//Time)}, \text{H(Sym\_2//Time)}, \ldots, \text{H(Sym\_n // Time)} \}_{\text{signCA-SC}} \quad (4)$$

At the present time, a typical SC [26] may store such a compliance certificate with an invalidation list with up to about 500 invalidated licenses. If the invalidation list becomes too big to be stored on the SC, then the certificate with the invalidation list can be stored, for instance, on a server in the network or on an optical storage medium.

As before, upon a user request for content access on a CoD, the SC must present its compliance certificate. After a mutual compliance check, the CoD sends PK[Sym//Rights//contentID] to the SC, which decrypts it and sends back the values Sym, Rights and contentID. But before the CoD uses Sym to decrypt the content to give access to the user, it calculates H(Sym//Time) and checks whether this value is in the invalidation list of the SC's compliance certificate or not. Only if it is not, the CoD proceeds with the handling of the access request.

23.4.2 Anonymous Licenses

In the previous section, the CP learns the association between the first and second user (i.e., between their public keys) when the license transfer is requested. If this is unwanted by the users, generic licenses in which a user identity is not specified can be used, as described below.

The generic license above (from now on referred to as *anonymous license*) is a license for a specified content with specified rights, but which is not associated with an identity (i.e., with a public key). Such a license can be issued by the CP for an anonymous user who pays for a given content with given rights as well as for the first user who requested the invalidation of his license, as described in the previous section. Since the license is not associated with any identity, it can be transferred (given, sold, etc.) to any other person. This person can later present the anonymous license to that CP and exchange it for a personalized license as given in (1). The latter can then be used for content access.

A security threat in this procedure is that users may copy the anonymous license and redeem multiple copies at different times. To prevent that, before the CP issues the anonymous license, a unique identifier is assigned to it. If this identifier is chosen by the CP, however, it will be able to link the public keys of the first and second user via that identifier. In order to prevent that, blind signatures [27] can be used, as described below.

A secret random identifier ID is created by the first user, who blinds this value (e.g., by multiplying ID by another randomly chosen value) and sends it to the CP. The user may also send a specification for new rights, NewRights, which are to be associated with the anonymous license, provided that NewRights allow less than the original rights. This possibility allows a user to give to another user a license with more restrictive rights than the original rights he had, if he so wishes.

For each combination of rights and content {Rights, contentID}, the CP has a unique pair of public-private keys. It is assumed here that the set of all rights is pre-specified consisting of, say, R rights and the set of all content has C items. So the CP must have $R{\times}C$ different public-private key pairs. Therefore, when the CP receives the data {Blind[ID], NewRights} from the first user, it signs Blind[ID] with the private key of

the combination {NewRights, contentID} and sends back the value $\{\text{Blind}[\text{ID}]\}_{\text{signed-NewRights-contentID}}$. The user then un-blinds the signed identifier to obtain $\{\text{ID}\}_{\text{signed-NewRights-contentID}}$. This protocol is depicted in Fig. 23.4 for content CD_1, and old and new rights as Rights_1 and Rights_2, respectively.

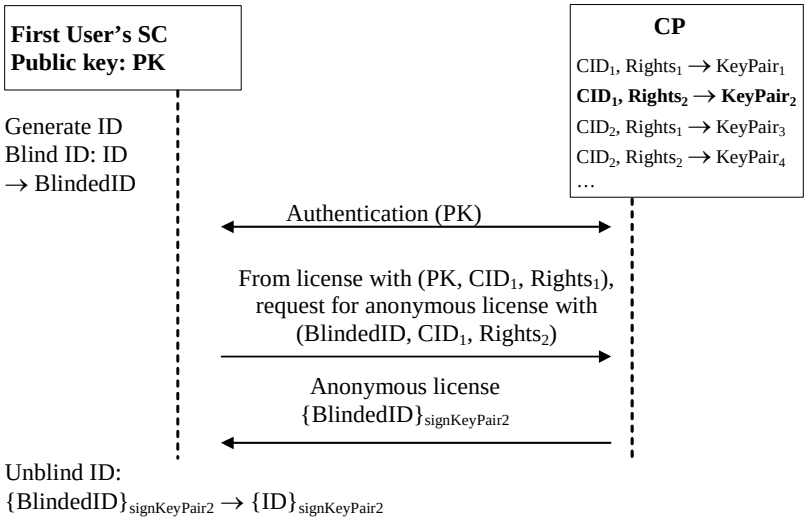

Fig. 23.4. Obtaining an anonymous license from the CP.

Next, the un-blinded value is sent to the second user together with the license specification {NewRights, contentID}. The second user can now contact the CP anonymously to obtain a personalized license. He authenticates himself and sends $\{\text{ID}\}_{\text{signed-NewRights-contentID}}$ and {NewRights, contentID} to the CP. The CP finds the correct key pair and checks its own signature in the value ID. If correct, the CP issues a personalized license to the second user, as given in (5), and sends it to the user.

$$\{\ \text{PK'}[\ \text{Sym'//NewRights//contentID}\ ],\ \text{H(NewRights//contentID)}\ \}_{\text{signCP}} \tag{5}$$

The protocol carried out between the second user and the CP is depicted in Fig. 23.5 for the example given in Fig. 23.4.

After the issuance of the license above, the value ID is entered by the CP into a list of used IDs. This prevents the personalized license request for an already redeemed anonymous license.

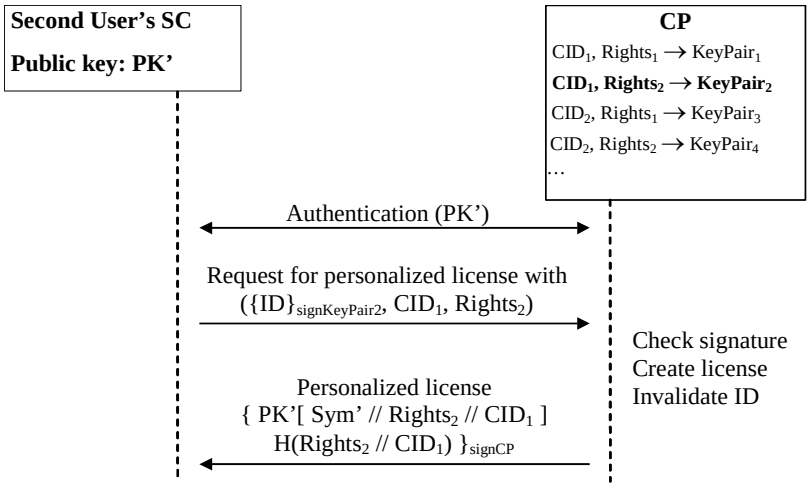

Fig. 23.5. Redeeming the anonymous license for a personalized one.

Note that the invalidation of the old license of the first user must be dealt with before the CP issues an anonymous license for that user. This allows an unlinkable transfer of licenses between users which is also secure. Another application relates to the business model of motivating users to buy a certain content, for instance, buy one, get a second one for free. The second license can be issued as an anonymous license which can be transferred to any person.

23.5 Discussion

A privacy-preserving DRM system is described, which protects users' privacy while preserving the system's security. Below, the privacy and security aspects of the system (basic as well as with extensions) are discussed.

User privacy is achieved in the DRM system by decoupling the user's real identity from his identifiers in the DRM system (i.e., PK and RAN). Concerning the relevant entities in the system, the following holds for a user with public key PK:

- The SCI learns the association (PK $\leftrightarrow$ PK*), but only if the SC is hacked.
- The CP learns the association (PK $\leftrightarrow$ (contentID, Rights, Sym)).
- The CA-SC learns the association (PK $\leftrightarrow$ RAN).
- The CoD learns the association (RAN $\leftrightarrow$ (contentID, Rights, Sym)).

It is therefore the case that, even by collusions of the parties above, the real identity of the user cannot be revealed since no parties know that identity.

The above statement regarding collusions is untrue only if an attacker can obtain user-related information from the CoD after a content access transaction happens. In this case, the associations

- (user's real identity ↔ RAN), and
- (user's real identity ↔ (contentID, Rights, Sym))

become known to him (if that information can be linked to the user's real identity). If collusion is not possible, however, the privacy damage is minimized: the attacker cannot learn the user's public key PK from the CoD, RAN changes periodically and only one piece of content is associated with the user's real identity. In this way, the attacker is prevented from creating a full log of the user's ownership of content and pattern of content usage.

To ensure the security of the DRM system, a compulsory mutual compliance check between SC and CoD must be carried out upon a content access transaction. The SC checks whether the CoD is compliant by means of a compliance certificate issued by the CA-CoD, and the CoD, in its turn, checks the SC for compliance, also by means of a compliance certificate. These certificates must be renewed often in order to ensure that the checks are up-to-date. The privacy of the user is preserved with the use of temporary pseudonyms (the RAN values) for the SC.

A privacy-enhancing extension of the system allows a user to further protect his privacy by purchasing content under different pseudonyms. In this case, the CP is unable to link all content bought by the same user, thus protecting his privacy. The various pseudonyms of the user must however be certified at a trusted authority (the CA) to guarantee the system security. Pseudonym certification guarantees that the pseudonyms

- are calculated from the original user's public key PK by the CA,
- are stored by the CA, connected with PK, and only revealed under certain conditions.

An additional privacy-enhancing extension of the system concerns the transfer of licenses between users. The solution proposed also guarantees the security of the DRM system, as explained below.

Security can be ensured with the invalidation of transferred licenses by means of the compliance certificate in (4). It includes the invalidation list with all invalidated licenses of a given SC. The frequent renewal of this certificate is important and done in the interest of both, the user and the DRM system, for the following reasons:

- for the user, it is done to minimize linkability, via the pseudonym RAN, of the user's content access requests to different content, and

- for the DRM system, it is done as a requirement of the CoD to check if the certificate (and therefore the license invalidation list) is too old via the value Time.

The user might not mind the linkability above, which would cause infrequent or no renewal actions on the part of the user. The renewal can be, however, forced as a requirement of the CoD, in order for that device to frequently get renewed values of invalidated licenses of PK.

The use of anonymous licenses in the license transfer process ensures user privacy. These licenses are anonymous (as they do not include any user identifier) and can be redeemed at the CP for real usable licenses. They must, however, include a unique identifier to be checked by the CP to prevent an anonymous license from being copied and redeemed multiple times. While guaranteeing system security, this unique identifier allows the CP to link the two users involved in the transfer. The use of blind signatures, however, ensures that this is not possible.

References

1. iTunes Music Store Downloads Top Half a Billion Songs, http://www.apple.com/pr/library/2005/jul/18itms.html
2. MSN Music Entertainment, http://music.msn.com/
3. RealPlayer Music Store, http://www.real.com/musicstore/
4. Sony Connect www.connect.com/
5. Rhapsody, http://www.rhapsody.com/
6. Napster, http://www.napster.com/
7. Windows Media 10 Series: Digital Rights Management (DRM), Internet Document, http://www.microsoft.com/windows/windowsmedia/drm/default.aspx
8. Apple's website, http://www.apple.com; see also http://en.wikipedia.org/wiki/FairPlay
9. Sony's Open MagicGate, Internet Document, http://www.sony.net/Products/OpenMG
10. Helix DRM from Real, Internet Document, http://www.realnetworks.com/products/drm
11. SmartRight, Internet Document, http://www.smartright.org
12. W. Jonker, J.-P. Linnartz, "Digital rights management in consumer electronics products", IEEE Signal Processing Magazine, Volume: 21, Issue: 2, 2004, pp. 82–91.
13. S.A.F.A. van den Heuvel, W. Jonker, F.L.A.J. Kamperman, P.J. Lenoir, "Secure Content Management in Authorised Domains", In Proceedings of the International Broadcasting Convention (IBC), 2002.
14. DVB-CPT, DVB-CPT Authorized Domain: Definition / Requirements, cpt-018r5, 2002
15. P. Koster, F. Kamperman, P. Lenoir and K. Vrielink, "Private Entertainment Domain: Concept and Design", Conf. on Communications

and Multimedia Security (CMS2005), September 19-21 2005, Salzburg, Austria.
16. J. Feigenbaum, M. J. Freedman, T. Sander and A. Shostack, "Privacy Engineering for Digital Rights Management Systems", In Proceedings of the ACM Workshop on Security and Privacy in Digital Rights Management, 2001.
17. Electronic Privacy Information Center (EPIC) – digital Rights Management and Privacy, Internet Document, http://www.epic.org/privacy/drm /default.html
18. D. Chaum, "Untraceable Electronic Mail, Return Addresses and Digital Pseudonyms", Communications of the ACM, vol. 24, no. 2, February 1981.
19. Anonymizer, http://www.anonymizer.com/
20. C. Conrado, F. Kamperman, G.J. Schrijen and W. Jonker, "Privacy in an Identity-based DRM System", Proceedings of the 14th International Workshop on Databases and Expert Systems Applications, Prague, Czech Republic, 2003.
21. A. Lysyanskaya, *Pseudonymous Systems*, Master's Thesis at the Massachusetts Institute of Technology, June 1999.
22. W. Diffie and M. Hellman, "New directions in cryptography", IEEE Transactions on Information Theory, 22, pp. 644–654, 1976.
23. T. Elgamal, "A Public-Key Cryptosystem and a Signature Scheme Based on Discrete Logarithms", IEEE Transactions on Information Theory, v. IT-31, n. 4, 1985, pp. 469–472 or CRYPTO 84, pp. 10–18, Springer-Verlag.
24. Digital Signature Standard (DSS), Internet Document, http://www.itl.nist.gov/fipspubs/fip186.htm
25. A. J. Menezes, P. C. van Oorschot, S. A. Vanstone, *Handbook of Applied Cryptography*, CRC Press, 1997.
26. Smart Card Basics, Internet Document, http://www.smartcardbasics.com
27. D. Chaum, "Blind signatures for untraceable payments", Advances in Cryptology: Proceedings of Crypto'82, Springer-Verlag, 1982.