A Personal DRM Scheme Based on Social Trust*

QIU Qin¹², TANG Zhi¹², LI Fenghua³⁴ and YU Yinyan¹²

(1. Institute of Computer Science and Technology, Peking University, Beijing 100871, China)  
(2. Beijing Key Laboratory of Internet Security Technology, Peking University, Beijing 100871, China)  
(3. Institute of Information Engineering, Chinese Academy of Sciences, Beijing 100029, China)  
(4. Department of Electronic Engineering, Beijing Electronic Science and Technology Institute, Beijing 100070, China)

Abstract — Existing commercial Digital rights management (DRM) schemes are not suitable for personal content protection because of centralized architecture and rigid constraints. To enable secure and flexible sharing of sensitive personal content, a new DRM scheme is proposed in this paper. Social trust between content sharers is modeled as computable concepts with DRM related contexts; based on the trust model, decentralized DRM architecture and scalable content sharing protocols are presented. Using the proposed DRM scheme, personal content owners can perform authentication and authorization without the intervention of Trusted authority (TA); by performing content sharing recommendations, authorized content users can conditionally have content shared with friends. Prototype implementation and simulation experiments indicate that the proposed scheme achieves satisfactory security and usability.

Key words — Digital rights management (DRM), Personal content, Social trust.

I. Introduction

Nowadays, more and more personal contents are created and shared by individuals for either casual or collaboration purposes. It is quite common that personal content owners create digital content with ubiquitous devices like smart phones and laptops, and have the content shared with friends or cooperators through Social Network Services, instant messengers or emails.

Some personal contents, such as contract drafts, unpublished original works, private photos and videos are involved with business benefits, copyrights, or privacy. The illegal disclosure of them often results in infringements on rights and interests of content owners. Digital rights management (DRM) provides protection in the whole life-cycle of digital content, and gives content owners varying degrees of control over how digital contents may be used¹²⁴. It is a desirable solution to protect sensitive personal contents.

However, most existing DRM systems, such as Microsoft Windows Media Rights Manager and InterTrust Rights—System are set up to preserve commercial profits of large and medium content providers¹⁴. They rely on trusted authority (TA) played by a license server and sometimes an external Certificate authority (CA) to record transaction information, authenticate content users and perform authorizations.

In recent years, a few DRM schemes for personal content protection turn up, but TA is still indispensable. Bhatt et al.¹ proposed a personal DRM manager for smart phones, which works on the assumption that there is CA to issue certificates for each device. Microsoft Information rights management (IRM)⁵ and Voltage SecureFile⁶ need a trusted server to authenticate content users and issue them licenses.

1. Limitations of existing DRM schemes

Some features of existing DRM schemes hinder their applications in personal content protection.

Limitation 1 The centralized system architecture causes expense and privacy problems for personal content owners.

Existing DRM systems set up centralized TA to authenticate and authorize content users. However, for content owners who are individuals and small organizations, building or employing centralized TA service is too costly. Even if there is a free service to use, content owners may be reluctant to have their secrets (such as the content encryption keys) and privacy (such as authorization records) in the charge of a third party. Therefore, it is fundamentally important for personal DRM systems to deploy decentralized architecture and enable autonomous authentication and authorization by content owners.

Limitation 2 Authorized content users can seldom share the content with others⁴. The constraint is reasonable in commercial DRM systems to reserve the benefits of content providers; however, in personal content sharing, the constraint is unreasonable and bothering. Personal content sharing usually happens among people with similar interests, goals or values, for either casual or collaboration purposes⁷. In many cases, content owners expect content users to re-share the content with friends to build stronger social ties or complete tasks better. For example, Alice may share her original work with Bob, hoping that Bob or Bob’s friends can help her improve the work. A desirable DRM

---

*Manuscript Received Nov. 2011; Accepted Dec. 2011. This work is supported by the National Natural Science Foundation of China (No.61170251), Key Program of Scientific and Technology Research of Ministry of Education (No.209156), and Beijing Natural Science Foundation (No.4102056).
scheme should help Alice ensure that the new content sharers introduced by Bob would use the content properly.

2. Our contributions

Social trust is a belief in the honesty, integrity and reliability of others[8]; it is the basic environmental factor of personal content sharing. Because the danger of being misquoted or discovering that the shared content has been used for underhanded or unsavory purposes is always there, before one shares important content, there is an assumed understanding of trust that the content will be used only for the good[9].

In this paper, we model social trust between content sharers, and propose a DRM scheme for personal content owners to share their contents with trusted ones autonomously and flexibly. By integrating social trust, the proposed scheme has the following advantages: (1) it is based on decentralized architecture; without the intervention of TA, a content owner can perform authentication and authorization autonomously. (2) With recommendations from content users, suitable new content users (those with similar interests or help in completing the task) can be introduced in a controllable manner.

II. Overview of our Scheme

1. System model

Fig.1 illustrates the model of our DRM scheme. Secure content sharing progresses between Content owner (CO) and Content user (CU), and no TA is involved in the system. CO issues herself Owner license (O-Lic), with which CO can perform any operation including package and authorization on all her contents. With User license (U-Lic) issued by CO, protected content can be securely decrypted and rendered by DRM Client software (DRM-CS) of CU.

In our scheme, the premise for authorization is that CO deems CU to be a trustworthy user, who will use the content properly. For example, CU is trusted not to plagiarize innovations in the protected original work and publish a similar work in advance. The general process of content sharing is as follows:

(1) CO establishes sharing trust with CU. If CO has knowledge about CU, the establishment can be directly completed by CO; if CU is unknown, CO can establish indirect trust with CU based on recommendations from others.

(2) After establishing sharing trust, CO issues U-Lic to CU, with which CU can not only use the content but also make sharing recommendations for her trusted entities to share the content.

2. DRM Client software (DRM-CS)

Any system user can act as CO or CU, and perform trust information management, authorization and content package/usage through DRM-CS. Fig.2 illustrates the main functional components of DRM-CS:

(1) Trust Manager manages trust-related information. Its main responsibilities are: (1) collecting and saving trust information from user inputs and recommendations of other system users; (2) evaluating the trustworthiness of target entities; (3) providing recommendations to others with local trust information.

(2) DRM Controller generates the client user’s public-private key pairs and content keys, implements cryptography-related operations, and sets right information based on trust information from Trust Manager; it also generates and interprets licenses.

(3) Content Package/Usage Tool is in charge of content package and usage.

Client reliability is foundation of all DRM schemes[3,4]. It enforces only legal operations on terminals without giving out any secret information. The reliability of our DRM-CS can be realized through pure software techniques. To resist tampers and reverse engineering attacks, software protection schemes like introspection, state inspection, and code obfuscation can be adopted[10]. For particular high security, hardware-aided techniques like Trusted Computing Platform[1] are available solutions.

III. The Underlying Trust Model

Trust is often deemed as a relationship that is reflexive (any entity trusts herself) and conditionally transitive (the fact that A trusts B and B trusts C does not indicate that A trusts C, unless certain conditions are satisfied). A trust model not only reflects existing trust relationships, but also builds new trust relationships with recommendation mechanism. There are three basic types of trust[8,11–13]:

(1) Direct trust reflects the trustor’s judgment on the trustworthiness of an acquainted entity, without intervention of third parties.

(2) Confidence of recommendation represents the trustor’s confidence in an entity to provide accurate recommendations.

(3) Indirect trust in an unknown entity is built through recommendations from those that have trust in the recommended one.
In this section, our trust model with DRM related contexts is presented. The notations in Table 1 are used throughout this paper.

### Table 1. Notations

<table>
<thead>
<tr>
<th>Notation</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>$UID_i$</td>
<td>$i$’s user identifier</td>
</tr>
<tr>
<td>$TD(i,j,context)$</td>
<td>$i$’s trust degree in $j$ under certain context</td>
</tr>
<tr>
<td>$CoR(i,j,context)$</td>
<td>$i$’s confidence degree in $j$’s recommendations under certain context</td>
</tr>
<tr>
<td>$VT_i(context)$</td>
<td>Validity Threshold set by $i$ under context</td>
</tr>
<tr>
<td>$PU_i,PR_i$</td>
<td>$i$’s public key; $i$’s private key</td>
</tr>
<tr>
<td>$fd(k●)$</td>
<td>key derivation function</td>
</tr>
<tr>
<td>$Sig_i$</td>
<td>$i$’s signature on message digest</td>
</tr>
<tr>
<td>$PEnc(pu●)$</td>
<td>asymmetric encryption with pu and</td>
</tr>
<tr>
<td>$PDec(pr●)$</td>
<td>asymmetric decryption with pr</td>
</tr>
<tr>
<td>$Enc(k●)$</td>
<td>symmetric encryption and decryption</td>
</tr>
<tr>
<td>$Dec(k●)$</td>
<td>with secret key $k$</td>
</tr>
</tbody>
</table>

### 1. Context-aware trust representation

Our trust model is context sensitive. For a trustor, her trust relationship towards a trustee is defined as below:

$$Trust(UID_{trustee}, context) = \{TD, CoR\}$$

where $context = (\text{trustCategory, trustConstraint})$.

In expression (1), $context$ is a feature vector providing background information including trust category and trust constraint. $TD$ is trustor’s trust (either direct trust or indirect trust) in the trustee under the specified context. $CoR$ is the trustee’s confidence in the trustor’s recommendations under the specified context. Being fuzzy logics, both $TD$ and $CoR$ are continuous variables in the interval of $[0,1]$. 0 indicates lowest degree of trust or confidence, while 1 indicates highest.

According to the contexts involved in the DRM system, we have two categories of trust: Key Trust and Sharing Trust.

**Key trust** ($KT$) is the trust in authenticity of the binding between the trustee and the claimed public key. It provides foundation for user authentication and secure communication. A trustor’s Key Trust towards a trustee can be described as $Trust(UID_{trustee}, (KT))$. Here trust constraint is set void.

**Sharing trust** ($ST$) is the trust in eligibility of the trustee to share the content. It provides foundation for user authorization. A trustor’s Sharing Trust towards a trustee can be described as $Trust(UID_{trustee}, (ST,CID))$. Here, trust constraint is a content identifier $CID$; it confines the range of contents that the trustee is trusted to share.

We use Validity threshold ($VT$) to map $TD$ and $CoR$ to valid or invalid states. $VT$ is a continuous variable in the open interval of $(0,1)$. It is adjustable by system users according specific contexts and security policies. For example, if Content Owner $S$ expects only very trustworthy entities to share a sensitive content $CID$, she can set a high value for $VT_i((ST,CID))$.

### 2. Trust propagation

A trustor’s trust relationship with other entities can be regarded as a directed graph, as seen in Fig.3. With recommendations from different recommenders, multiple recommendation paths connecting the trustor to the target entity are built. The trustor propagates trust along all paths to evaluate the trust degree in the target entity.

In expression (1), $TD$ is a continuous variable in the interval of $[0,1]$ where 0 indicates lowest degree of trust or confidence, while 1 indicates highest.

According to the contexts involved in the DRM system, we have two categories of trust: Key Trust and Sharing Trust.

**Key trust** ($KT$) is the trust in authenticity of the binding between the trustee and the claimed public key. It provides foundation for user authentication and secure communication. A trustor’s Key Trust towards a trustee can be described as $Trust(UID_{trustee}, (KT))$. Here trust constraint is set void.

**Sharing trust** ($ST$) is the trust in eligibility of the trustee to share the content. It provides foundation for user authorization. A trustor’s Sharing Trust towards a trustee can be described as $Trust(UID_{trustee}, (ST,CID))$. Here, trust constraint is a content identifier $CID$; it confines the range of contents that the trustee is trusted to share.

We use Validity threshold ($VT$) to map $TD$ and $CoR$ to valid or invalid states. $VT$ is a continuous variable in the open interval of $(0,1)$. It is adjustable by system users according specific contexts and security policies. For example, if Content Owner $S$ expects only very trustworthy entities to share a sensitive content $CID$, she can set a high value for $VT_i((ST,CID))$.

### IV. Operation Process and Related Protocols

In this section, we describe how our trust-based scheme works to secure personal content sharing. The whole process consists of four steps, as illustrated in Fig.5.
Step 1 Initialization

Step 1.1 To begin with, system user S generates a random public-private key pair \{PU\_S, PR\_S\} with her DRM-CS. PR\_S is stored in the form of ciphertext.

Step 1.2 S establishes Key trust (KT) and exchanges KT with D and saves \{KT\_D\}. After the deadline \(\ell\), S’s DRM-CS verifies \(\ell\) at any time in any way.

Step 1.3 S’s DRM-CS generates O-Lic \(LO(S)\), which contains the cipher of a random master key \(MK\_S\). With \(LO(S)\), S fully controls all her contents.

\[ S : LO(S) = \{UID\_S, PE\_PU(PU\_S, MK\_S), RightsInfo, Sigs\} \]

Step 2 Content package

To protect some content M, S’s DRM-CS generates a unique content identifier CID, collects MK\_S from \(LO(S)\), and then derives content encryption key CEK from MK\_S and CID with a function satisfying one-way and randomness\[^{15}\]. Next, S’s DRM-CS encrypts M and generates content package CP. CP can be distributed to CU at any time in any way.

\[ S : CEK = kd(CID, MK\_S) \]
\[ C = Enc(CEK, M) \]
\[ CP = \{CID, UIDs, C, Sigs\} \]

Step 3 CU authorization

When content sharing happens, S first establishes Sharing trust (ST) with CU, and then performs authorization for them.

Case 3a CU is directly trusted

For a direct trustee R, S sets trust constraint CID, as well as \(TD(S, R, (ST, CID))\) and \(CoR(S, R, (ST, CID))\).

If \(TD(S, R, (ST, CID)) > VT_{S}(ST, CID)\), S deems R as an eligible content sharer and generates U-Lic \(LU(R)\) for R. To preserve R’s privacy, authorization information is encrypted with system default key SysKey as \(\ell\).

\[ S : \ell = Enc(SysKey, \{CID, RightsInfo\}) \]
\[ S \to R : LU(R) = \{UID\_S, UIDs, \ell, PE\_PU(CEK), Sigs\} \]

Case 3b CU is indirectly trusted

Authorized CU may want to re-share content CID with an unauthorized entity D, and sends re-sharing request to S. If D is unknown to S, S sends Sharing recommendation request (SRR) to \(R_{k}\) (where \(k = 1, 2, \cdots\)) who are authorized CU of CID. SRR contains the recommendation deadline \(\tau\), and a random number \(\gamma\) to prevent message replay.

\[ S : \alpha = Enc(SysKey, \{CID, UID\_D\}) \]
\[ S \to R_{k} : SRR = \{UID\_S, \alpha, \tau, \gamma, Sigs\} \]

If having direct ST in D, \(R_{k}\) returns S a Recommendation Certificate \(RecCert(R_{k})\) with trust information encrypted to protect privacy.

\[ R_{k} : \beta_{k} = Enc(SysKey, \{CID, UID\_D, TD(R_{k}, D, (ST, CID))\}) \]
\[ R_{k} \to S : RecCert(R_{k}) = \{UID\_R_{k}, UID\_D, \beta_{k}, \gamma, Sigs\} \]

After the deadline \(\tau\), S’s DRM-CS verifies \(\gamma\) and recommenders’ signatures in received recommendations, and then propagates \(TD(S, D, (ST, CID))\). If the result is larger than \(VT_{S}(ST, CID)\), S deems D to be an eligible CU of CID, and generates U-Lic \(LU(D) = \{UID\_S, UID\_D, \ell, PE\_PU(CEK), Sigs\}\) for D.

Step 4 Content usage

To use content, D’s DRM-CS first associates CP with \(LU(D)\) by checking whether CID in \(LU(D)\) and that in CP are identical, and then ensures that the issuer identifier in \(LU(D)\) and the owner identifier in CP are the same. After successful verification, D’s DRM-CS collects CEK from \(LU(D)\) to decrypt the content cipher in CP.

V. Security Analysis

1. Robustness of the trust model

Illegal CU may be introduced in two ways: (1) Because of subjective faults, a trustor overvalues trust degrees in trustees, causing that trust degrees in some untrustworthy entities turn larger than VT mistakenly; (2) some rogue recommenders may provide unfair positive recommendations for untrustworthy entities individually or collusively. We carried experiments simulating the above ways to test the robustness of our trust model.

Simulation 1-Trust overvaluation

we simulated that trustors overvalue the trust degrees in all their trustees with random scales within an overvaluing range. Experiments were carried out in random trust networks with 100 trustees.

Simulation 2-Unfair positive recommendations

we carried out experiments in random trust networks with 100 recommenders and 100 recommendees. When unfair recommendation attacks happened, random rogue recommenders assigned the highest trust degree (i.e. 1) to all they recommend.

Experiment results

The experiment results of Simulation 1 and Simulation 2 are shown in Fig.6 and Fig.7 respectively. The results indicate that: (1) our trust model achieves...
satisfactory robustness; the proportions of illegal CU are in very low levels; (2) setting a proper value for VT helps impede the appearance of illegal CU.

![Image](46x739 to 557x739)

**Fig. 6.** The proportion of illegal CU caused by trust overvaluation

![Image](75x568 to 263x686)

**Fig. 7.** The proportion of illegal CU caused by unfair positive recommendations

### 2. Security of content sharing protocols

With the protocols described in Section IV, our scheme achieves following properties:

**Property 1** Only system users that are authorized by CO can decrypt and use the content.

The reason is twofold: (1) Because CEK (or its origin MK) in a license is encrypted with the private key of the target system user (either CO or CU), only those system users who have been issued a license can use the content. (2) Before using the content, compliant system users verify whether the owner of the content package and the issuer of the license are identical. Thus, only the corresponding CO can issue valid licenses.

**Property 2** The privacy of system users can be hardly disclosed at a large scale.

Two kinds of privacy information are involved in our scheme: authorization information in U-Lic and trust information in recommendation certificates. They are encrypted with SysKey and can only be decrypted by the DRM-CS of the target receiver after successful verification. Nobody else except the message sender knows the plaintext of the privacy information.

By requesting only one recommender for recommendations, a malicious trustor may infer the recommender’s trust information from the result of trust propagation. However, such method is low-efficient and troublesome. In a macroscopical view, it can hardly cause privacy concerns on a large scale of system users.

### VI. Conclusions

In this paper, we model social trust among content sharers, and propose a DRM scheme to protect personal content sharing. To our best knowledge, we are the first to integrate social trust into DRM application\[8,11–13\]. A comparison of our work with related solutions for personal content protection is shown in Table 2. The merits of our scheme include: (1) It is TA independent, which eliminates the cost and privacy problems in existing centralized DRM systems; (2) It supports recommended content sharing, which enables more flexible sharing experiences.

Our scheme can be used to secure private information sharing, business collaborations among small-scale organizations, and original work appreciation before the work is issued for sale. A prototype of DRM-CS has been developed, which is composed of a desktop manager and a plug-in in file readers. Content-related operations, including package, authorization, and usage, are performed by the plug-in, while trust information is managed by the desktop manager. For a plaintext file with the size of 100K bytes, the decryption time of its cipher is 0.61 milliseconds (on PC with Pentium D CPU 3.00GHz, and 1.00GB RAM) with U-Lic.

### Table 2. Comparisons of personal content protection solutions

<table>
<thead>
<tr>
<th></th>
<th>Refs.[2,5,6]</th>
<th>Ref.[10]</th>
<th>Our work</th>
</tr>
</thead>
<tbody>
<tr>
<td>Persistent protection</td>
<td>Yes</td>
<td>No</td>
<td>Yes</td>
</tr>
<tr>
<td>TA independence</td>
<td>No</td>
<td>Yes</td>
<td>Yes</td>
</tr>
<tr>
<td>Support recommended</td>
<td>No</td>
<td>No</td>
<td>Yes</td>
</tr>
</tbody>
</table>

### References


QIU Qin was born in Shangrao, Jiangxi Province, China in 1986. She received B.S. degree in computer science and technology from Beijing Normal University in 2007. She is currently pursuing Ph.D. degree in computer application in Peking University. Her research interests include digital rights management and information security. (Email: qiuqin@icst.pku.edu.cn)

TANG Zhi was born in Zhejiang Province, China in 1965. He received B.S. degree in radio physics from Peking University in 1987, and received M.S. degree and Ph.D. degree in computer application from Peking University in 1990 and 1995 respectively. He is a professor and doctoral supervisor in Institute of Computer Science and Technology of Peking University. His research interests include document processing and digital rights management. (Email: tangzhi@icst.pku.edu.cn)

LI Fenghua was born in Xishui, Hubei Province, China in 1966. He received B.S. degree, M.S. degree, and Ph.D. degree in computer software and computer systems architecture from Xidian University, China, in 1987, 1990, and 2009 respectively. He had been a lecturer in Xidian University from 1992 to 1994. Since 1994 he has been with Beijing Electronic Science and Technology Institute as a lecturer, associate professor, professor, and doctoral supervisor. His research interests include network security, system security & evaluation and trusted computation. (Email: lfh@besti.edu.cn)

YU Yinyan was born in Zhejiang Province, China in 1976. She received B.S. degree in computational mathematics from Nanjing University of Science and Technology in 1998, M.S. degree in applied mathematics from Chinese Academy of Sciences in 2001, and Ph.D. degree in computer application from Peking University in 2005. She is a senior engineer in Institute of Computer Science and Technology of Peking University. Her research interests include digital rights management and information security. (Email: yuyinyan@icst.pku.edu.cn)