A Trust-based DRM Scheme for Content Sharing in an Open Environment

Qin Qiu, Zhi Tang, Yinyan Yu
Institute of Computer Science and Technology
Peking University
Beijing, China
e-mail: {qiuqin, tangzhi, yuyinyan}@pku.edu.cn

Abstract—**Interpersonal or inter-organizational content sharing is a popular activity for casual or cooperation purposes. On one hand, content sharing is turning more and more open for better outcome or stronger influence; on the other hand, it is important to protect shared sensitive content from being misused or disclosed to malicious users. To secure content sharing in open environment, this paper proposes a DRM scheme built upon a trust model. With the proposed scheme, secure content sharing is open to all trusted content users, and user authentication and authorization can be performed autonomously by content owners. Experiments and comparisons indicate that the proposed scheme achieves satisfactory security and usability.**

Keywords-DRM; trust model; content sharing

I. Introduction

With the popularization of electronic devices and the development of Internet, lots of digital content are created and shared among individuals or organizations for casual or cooperation purposes. Examples of such open sharing include:

- Alice creates an original work and shares it with her friend Bob, expecting Bob or Bob's friends to offer some advices for improvement;
- Organization A cooperates with partner organization B on an innovation, and allows other unknown but eligible organizations to join for better outcome.

On one hand, the content owner may want the content to be shared with more users for better cooperation outcome or enlarge the influence; on the other hand, to preserve rights or interests, the content owner needs to have control on who and how to use the content.

Digital Rights Management (DRM), which achieves persistent content protection in the whole life-cycle of digital content and controls how digital content may be used [1], is a desirable solution to protect the shared content. However, existing DRM schemes serve for closed systems and depend on Trusted Authority (TA), who has priori-knowledge of all content users, to authenticate users and issue licenses [1-4]. The dependence on TA hinders existing DRM schemes from being applied into secure content sharing in open environment: firstly, it is impossible for TA to supervise all potential content users in an open environment; secondly, content owners may be reluctant to have their authorization information in the charge of a third party for privacy concerns. Therefore, it is important to enable autonomous rights management by content owners and provide a mechanism for content owners to evaluate the eligibility of potential content users in an open environment.

Social trust is a belief in the honesty, integrity and reliability of others; it is the basic environmental factor of content sharing [5]. 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 [6]. For example, Alice shares her original work with Bob on condition that Bob is trusted not to plagiarize innovations in the work and publish a similar work in advance. Content sharing in an open environment, which assumes that anyone may be a potential participant, is inherently a social activity. Establishing of trust in this context inevitably requires some form of social computing supported by a trust model [7].

To achieve secure content sharing in an open environment, we model social trust between content owners and content users, and propose a decentralized DRM scheme in this paper. The trust model enables content sharing with unknown content users, and eliminates the necessity of TA; authentication and authorization are performed autonomously by content owners. To reduce interaction and authorization overheads on content owners, contents are organized into groups, and a batch authorization method based on key derivation mechanism is integrated into the DRM scheme.

II. Related Work

A. DRM Schemes

Most existing DRM systems, such as Microsoft Windows Media Rights Manager and InterTrust Rights|System are set up to preserve the commercial profits of content providers. In those systems, License Server that is trusted by content providers is indispensable; it records transaction information and issues content users licenses for requested content [1-3]. Sometimes external Certificate Authority (CA) is also needed for identity authentication and certificate issuance [4]. Content users are only consumers of the protected content.

A few DRM schemes have been presented to secure content sharing; however, those systems are for closed systems where all content users are pre-known to content

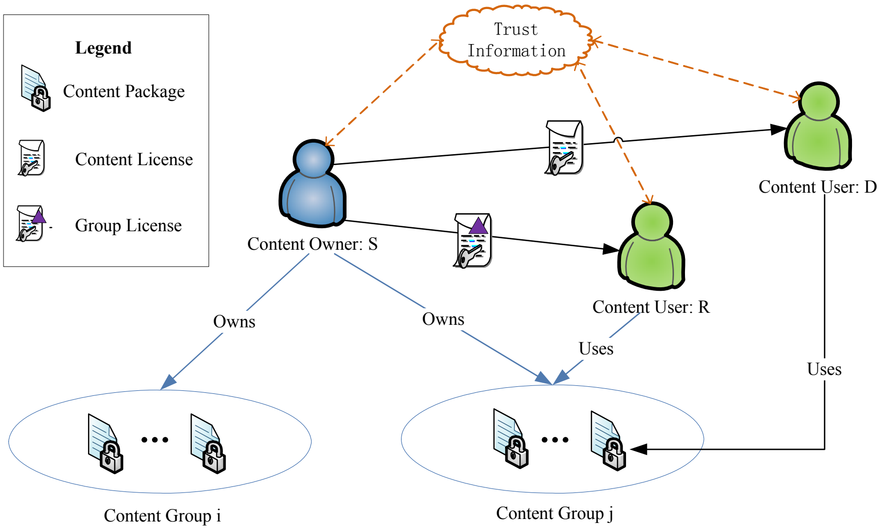


Figure 1. The model of our DRM system

owners and the system. Microsoft IRM [8] and Voltage SecureFile [9] need a trusted server for user enrollment, authentication and license issue according to content encryption keys and permission lists from content owners. The problem is that the server is able, though not bound, to decrypt content owners' secret contents and grasp the relations among content owners, content users and shared contents. Bhatt et al. [3] proposed a personal DRM manager for content sharing between smart phones, which works on the assumption that each participating smart phone holds a certificate issued by CA. Feng [10] proposed a decentralized copy protection solution, but it requests that there is pre-established trust relationship between content owners and content users.

In short, there is hardly any DRM solution for secure content sharing in an open environment.

B. *Trust Model*

Trust modeling was first proposed by Marsh [5] to assist decision making in distributed artificial intelligence systems. Till today, many trust models have been presented in areas of public key authentication [11], ubiquitous computing [12], and distributed network [13, 17]. In trust models, trust is generally regarded to be non-symmetrical (the fact that A trusts B does not indicate that B trusts A), 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) [5, 12].

There are three basic types of trust in a trust model [11-17]:

- *Direct trust* reflects the trustor's judgment on the trustworthiness of an acquainted entity, without intervention of third parties.
- *Confidence of recommendation* represents the trustor's confidence in an entity to provide accurate recommendations.
- *Indirect trust* in an unknown entity is built through recommendations from those that have trust in the recommended one. By performing some evaluation on the recommendations, the trustor can make judgment on the trustworthiness of the recommended entity.

Trust context is considered in some trust models. Abdul-Rahman [14] uses trust category to express particular semantic of trust, so that the model can be used in different applications. Ray [16] uses a set of keywords with equivalent semantics to represent context, so that trust relationships in same or similar contexts can be compared.

To our best knowledge, no trust model has been presented or applied in the area of DRM. Some researchers proposed all-purpose trust models [14, 17], but they are not so suitable to be directly used in our DRM scheme. In Rahman's trust model [14], users can only claim what the trust is about in the one-dimension context, not able to clarify more complex information like trust conditions or constraints; Liu's trust model [17] allows users to self-define trust contexts through XML schema, which is cumbersome and difficult to adopt. To be applied in DRM, a trust model has to be aware of context information related with user authentication and content authorization. We present a tailored trust model with contexts about trust types, constraints, and objects in Section IV.

III. System Model

In our scheme, secure content sharing progresses between Content Owner and Content User in client-to-client communication model and no TA is involved in the system. DRM agent of Content Owner and Content User manages trust information, content information and authorization information.

Fig. 1 shows the model of our scheme. As the existence of social trust is the premise of authorization, the general process of content sharing is as follows:

TABLE I. NOTATIONS

Notation	Description
//	or, connecting alternative items
UID_i	user identifier
TD(i,j,c)	user i's trust degree in user j under context c
CoR(i,j,c)	user i's confidence degree in user j's recommendations under context c
$VT_i(c)$	Validity Threshold set by user i under context c
PU_i; PR_i	public key; private key
Kd(•,•)	key derivation function
Sig_i	signature on message digest
PEnc(pu,•)	asymmetric encryption with key pu
Enc(k,•)	symmetric encryption with key k
H(•)	Hash function

(1). Content Owner establishes Sharing Trust with Content User. If the owner has knowledge about the user, the establishment can be directly completed by the owner; if the user is unknown, the owner can establish indirect trust with the user based on recommendations from other Content Users.

(2). After establishing Sharing Trust, Content Owner issues a license to Content User. In batch authorization mode, the license is Group License, with which Content User is able to use any content that is or will be categorized into the content group.

The notations in Table I are used throughout this paper.

IV. THE UNDERLYING TRUST MODEL

To be applied in open sharing environment, a distributed trust model is built in DRM agent. In this section, we describe how we model social trust between system users for DRM application.

A. *Trust Representation and Decision*

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

$$\text{Trust}(\text{UID}_{\text{trustee}}, \text{context})=\{\text{TD}, \text{CoR}\}; \qquad (1)$$
$$\text{context}=\langle\text{trustCategory}, \text{trustConstraint}\rangle.$$

- $UID_{trustee}$ is the user identifier of the trustee.
- context is a feature vector providing background information including trust category and trust constraint. While trustCategory is used to discriminate different kinds of trust involved in DRM application, trustConstraint limits the range that the trust is valid in.
- TD is trustor's trust (either direct trust or indirect trust) degree in the trustee under the specified context. CoR is the trustor's confidence degree in the trustee'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.

```
Procedure: TrustPro(source, dest, type)

rslt←0, j←0.
if there is a direct trust path from source to dest then
  if type=1 then
    rslt←TD(source, dest)
  else if type=2 then
    rslt←CoR(source, dest)
  end if
else n←the number of recommendation paths from source to dest
  if n>=1 then
    for every recommendation path i<=n do
      source finds recommender R_i that has direct trust path to dest
      CoR(source, R_i)← TrustPro(source, R_i, 2)
      if CoR(source, R_i)>VT then
        j←j+1;
        if type=1 then
          rslt_j←CoR(source, R_i)*TD(R_i, dest)      ----(2)
        else if type=2 then
          rslt_j←CoR(source, R_i)*CoR(R_i, dest)     ----(3)
        end if
      end if
    end for
    N←j
    rslt←average of rslt_k, where k=1,2,⋯,N              ----(4)
  end if
end if
return rslt
```

Figure 2. Procedure for trust propagation

- 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. A trustor's Key Trust towards a trustee can be described as $\text{Trust}(\text{UID}_{\text{trustee}},\langle\text{KT}\rangle)$. Here, trustConstraint 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 $\text{Trust}(\text{UID}_{\text{trustee}},\langle\text{ST,CID//GID}\rangle)$. Here, trustConstraint is a content identifier CID or a content group identifier GID; 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 to specific contexts and security policies. For example, if Content Owner S expects only very trustworthy entities to share sensitive content in group GID, she can set a high value for $VT_S(\langle ST,GID\rangle)$.

B. *Trust Propagation*

A trustor's trust relationship with other entities can be regarded as a directed graph. 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 [5, 12-17].

The procedure of our trust propagation is in Fig. 2. There are three input parameters, and the procedure outputs the trust propagation result. The input parameter *source* is the

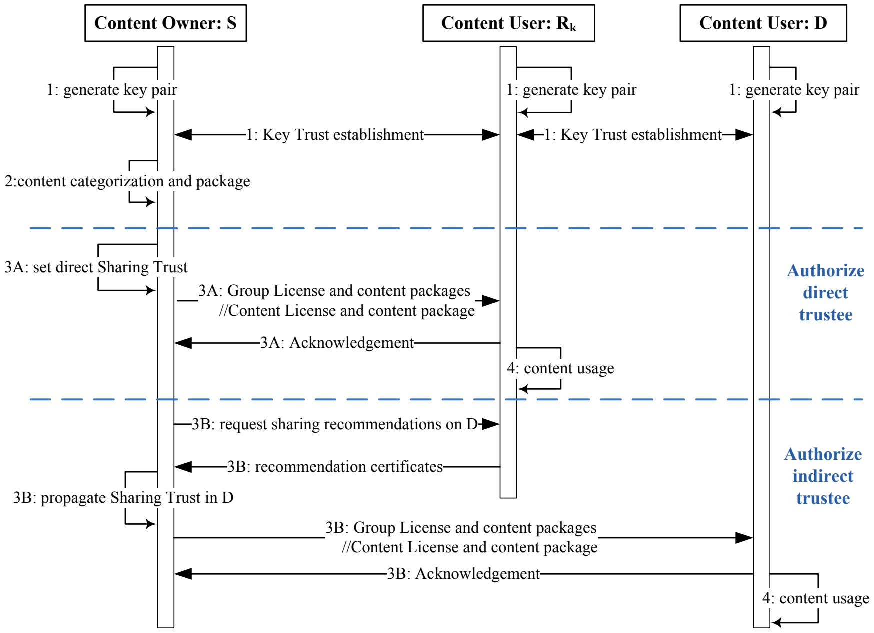

Content Owner: S
Content User: R_k
Content User: D
1: generate key pair
1: generate key pair
1: generate key pair
1: Key Trust establishment
1: Key Trust establishment
2:content categorization and package
3A: set direct Sharing Trust
3A: Group License and content packages
//Content License and content package
3A: Acknowledgement
4: content usage
Authorize direct trustee
3B: request sharing recommendations on D
3B: recommendation certificates
3B: propagate Sharing Trust in D
3B: Group License and content packages
//Content License and content package
3B: Acknowledgement
4: content usage
Authorize indirect trustee

Figure 3. Overview of operations and protocols

trustor's user identity, and *dest* is the target trustee's user identity. When input parameter *type* is set "1", the procedure outputs TD(*source*, *dest)*; when *type* is set "2", the procedure outputs CoR(*source*, *dest*).

In Fig. 2, we use expression (2), (3) and (4) for trust propagation because they conform to both Weighted Average Operator in D-S theory and Consensus Operator in Subjective Logic [15, 18]. To avoid the problem of opinion dependence [15], the procedure only considers the direct trust path and ignores all recommendation paths when an entity has direct trust in the target trustee. With the maximal length of recommendation paths limited with a reasonable constant, the complexity of the procedure is O(n), where n is the scale of valid recommenders.

V. Operations and Protocols

This section describes how our trust based scheme works in enabling secure content sharing in open environment. As illustrated in Fig. 3, the whole process consists of four phases: initialization, content categorization and package, content authorization, and content usage.

A. Phase 1-Initialization

To begin with, system user S sets a unique user identifier UID$_S$ through her DRM agent. The agent generates a public-private key pair {PU$_S$,PR$_S$ } for S.

Next, S establishes Key Trust and exchanges public keys with others. Firstly, with secure communication or auxiliary verification methods, S gets legal public keys of some friends R$_i$, i=1,2,⋯, and establishes direct Key Trust with them. When S needs the public key of some unknown system user D, S requests friends for recommendations. If a friend R$_i$, $\forall i$, has direct Key Trust in D, R$_i$ returns S a recommendation containing UID$_D$, PU$_D$ and TD(R$_i$,D,<KT>); otherwise, R$_i$ forwards the request to the next hop. Finally, S's DRM agent performs trust propagation on all the received recommendations. If the result is a valid trust value, S successfully builds Key Trust with D and saves PU$_D$.

B. Phase 2-Content Categorization and Package

Content Owner S sets up some content groups, each of which is assigned a unique group identifier GID and a random secret key GK . All contents in a content group have some identical properties, and target same Content Users.

When S needs to protect some content, S firstly categorizes it into a content group GID, assigns it a content identifier CID, and then derives content encryption key CEK from GK and CID with a key derivation function that satisfies one-way and randomness [19,20]. Next, S encrypts content plaintext M, and packages the cipher text with GID, CID and signature. CP can be distributed to Content Users in any way at any time.

S: CEK=kd(CID, GK)
C=Enc (CEK, M)
CP= {UID$_S$, GID, CID, C, Sig$_S$}

C. Phase 3-Content Authorization

In an open environment, content sharing may happen not only between friends, but also between strangers. In this phase, Content Owner first establishes direct or indirect Sharing Trust with Content Users, and then performs authorization for them. According to the authorization mode, there are two kinds of licenses:

- Group Licenses are issued for Content Users to use all contents that are or will be categorized to the corresponding content group.
- Content Licenses are issued for Content Users to use only the prescribed content.

We first describe authorization for directly trusted Content Users, and then authorization for indirectly trusted Content Users.

1) Direct Trust Establishment and Authorization.

Suppose there is direct Sharing Trust from Content Owner S to Content User R, and they have stable content sharing relation. S sets GID as the range of contents that is ready for R to use, TD (S,R,<ST,GID>) as the trust degree in R to share the contents in group GID, and CoR (S,R,<ST,GID>) as the confidence degree in R to recommend other Content Users to share one or more contents in the group GID.

If TD(S,R,<ST,GID>) > VT$_S$(<ST,GID>), S deems R as an eligible content sharer of content group GID, and generates Group License L$_G$(R) for R. To preserve R's privacy, authorization information is encrypted with system default key SysKey as ℓ.

S: ℓ=Enc(SysKey,{GID,RightsInfo})

S→R: L$_G$(R)={UID$_S$,UID$_R$, ℓ,PEnc(PU$_R$,GK),Sig$_S$}

After receiving L$_G$(R), R collects ℓ and PEnc(PU$_R$,GK) from it, and then returns S an acknowledgement message AM.

R→S: AM={UID$_R$,UID$_S$,H(ℓ,PEnc(PU$_R$,GK)),Sig$_R$}

2) Indirect Trust Establishment and Authorization.

Suppose Content User D, who is unknown to S, wants to share content CID. As CID belongs to content group GID and S has no direct Sharing Trust in D, S sends Sharing Recommendation Request (SRR) to R$_k$ (k=1,2,…) who are authorized Content Users of GID or CID. SRR contains the recommendation deadline τ, and a random number γ to prevent message replay. It should be noted that Content Owner only asks authorized Content Users for sharing recommendations, because only authorized Content Users can make proper judgment about whether the content can be shared by a candidate user.

S: α =Enc(SysKey,{CID,UID$_D$})

S→Rk: SRR={UID$_S$, α, τ, γ, Sig$_S$}

If having direct Sharing Trust in D, R$_k$ returns S a Recommendation Certificate RecCert(R$_k$) with trust information encrypted to protect privacy.

R$_k$: β_k =Enc(SysKey,{CID,UID$_D$,TD(R$_k$,D,<ST,CID>)})

R$_k$→S: RecCert(R$_k$)={ UID$_{R_k}$,UID$_S$, β_k, γ, Sig$_{R_k}$ }

After the deadline τ, S's DRM agent verifies γ and recommenders' signatures in received recommendation certificates, 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 Content User of CID, and generates Content License for D.

S: ℓ'=Enc(SysKey,{CID,RightsInfo'})

S→D: $\mathrm{L_C(D)} = \{\mathrm{UID_S}, \mathrm{UID_D}, \ell', \mathrm{PEnc(PU_{D,}CEK)}, \mathrm{Sig_S}\}$

After receiving L$_C$(D), D collects ℓ' and PEnc(PU$_D$,CEK) from it, and then returns S an acknowledgement message AM'.

D→S: AM'={ UID$_D$,UID$_S$,H(ℓ',PEnc(PU$_D$,CEK)),Sig$_D$ }

In another case, if D wants to share all contents belonging to GID, S propagates Shaing Trust in D with trust constraint GID; if the trust establishment is successful, S issues D a group license.

D. Phase 4-Content Usage

1) Usage with Group License. R's DRM agent first associates CP with L$_G$(R) by checking whether GID in L$_G$(R) and that in CP are identical, and then ensures that the issuer identifier in L$_G$(R) and the owner identifier in CP are the same. After successful verification, R's DRM agent derives CEK with GK collected from L$_G$(R) and CID collected from CP, and then decrypts the content cipher in CP.

2) Usage with Content License. D's DRM agent associates CP with L$_C$(D) according to CID and the owner identifier; next, D's DRM agent directly obtains CEK by decrypting its cipher in L$_C$(D), and then decrypts the content cipher in CP; finally, the agent manages content usage according to rights information in L$_C$(D).

E. Revocation of Group License

Suppose R is a frequent Content User of contents in group GID owned by S, and has been issued a group license L$_G$(R). When S wants to revoke L$_G$(R), so that R cannot use contents categorized into group GID after the revocation, there are two methods.

1) Group Alteration: S builds a new content group GID' that is associated with GID, and issues Group Licenses corresponding to GID' to valid Content Users. New contents are categorized to GID' instead of GID.

2) Key Update: S updates the group key of GID to be GK', and issues an updated Group License containing the cipher of both GK and GK' to valid Content Users.

VI. Security Analysis

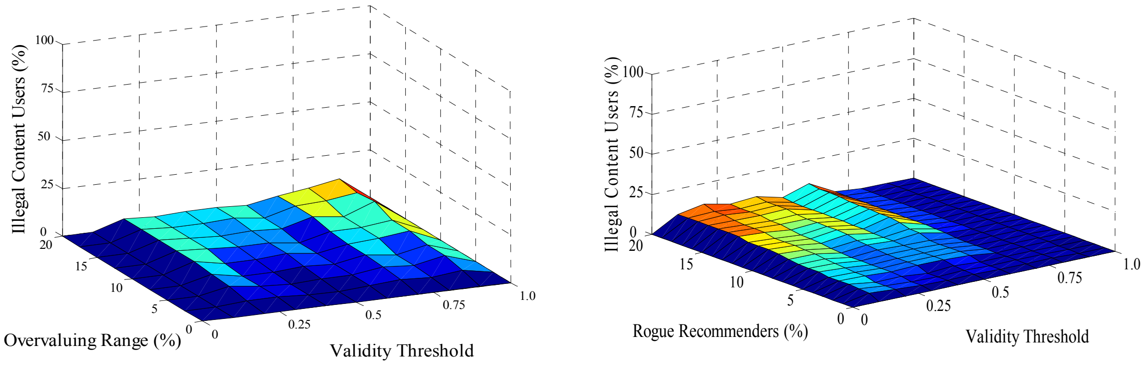
A. Robustness of the Trust Model

Illegal Content Users may be introduced in two ways: (1) a trustor overvalues trust degrees in trustees out of subjective faults, causing that trust degrees in some untrustworthy entities turn larger than VT mistakenly; (2) some rogue recommenders provide unfair positive recommendations for untrustworthy entities individually or collusively.

To test the robustness of our trust model, we simulated the above two ways in random trust networks; VT was set from 0 to 1 to observe its effects to the result. Shown in Fig. 4, the simulation results indicate that: (1) the proportions of illegal Content Users are in very low levels, and our trust model achieves satisfactory robustness; (2) setting a proper value for VT helps impede the appearance of illegal Content Users.

B. Proctection of User Privacy

There are mainly two kinds of privacy information involved in our scheme: authorization information in licenses and trust information in recommendation certificates. Both of them are encrypted with system keys and can only be decrypted by the DRM agent of the target receiver. Nobody



(a) (b)

Figure 4. The proportion of illegal Content Users caused by: (a) trust overvaluation simulations where trustors overvalue trust degrees in all their trustees with random scales within a range from 0% to 20%; (b) unfair positive recommendations simulation where random rogue recommenders (occupying from 0% to 20% of all recommenders) assigned the highest trust degree (i.e. 1) to all they recommend

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. It can hardly cause privacy concerns on a large scale of system users.

VII. Prototype Implementation

We have developed a prototype system of the proposed scheme to protect Microsoft Office Word 2007 documents and enable secure document sharing in an open lab. The system is composed of a desktop manager and a Microsoft Office Word 2007 plug-in. Through the desktop manager, users can manage personal information and trust relationships; through the plug-in, users can perform content protection and usage operations. The main User Interface (UI) of the prototype system is shown in Fig. 5.

In the prototype system, content packages, licenses and recommendation certificates are all described in XML files. The file size of a license is about 393 bytes. For a plaintext document with the size of 1124 kilobytes (KB), the encryption time is 35.692 milliseconds by Content Owner, and the decryption time of the corresponding content package is 3.392 milliseconds by Content User with a content license (tests were carried out on a PC with Pentium D CPU, 3.00GHz and 1.00GB RAM).

VIII. Conclusion

In this paper, we propose a DRM scheme to secure content sharing among those with direct or indirect social ties. A comparison of our scheme with related solutions is shown in Table II. Based on the social trust model, our scheme has the following advantages:

(1). It is independent of TA; authentication and authorization are performed by Content Owner autonomously.

(2). Unknown content users in open environment may participate in content sharing according to the result of trust evaluation.

(3). According to the constraint information of Sharing Trust, Content Owner can perform either content authorization or group authorization, which achieves a good balance between security and authorization efficiency.

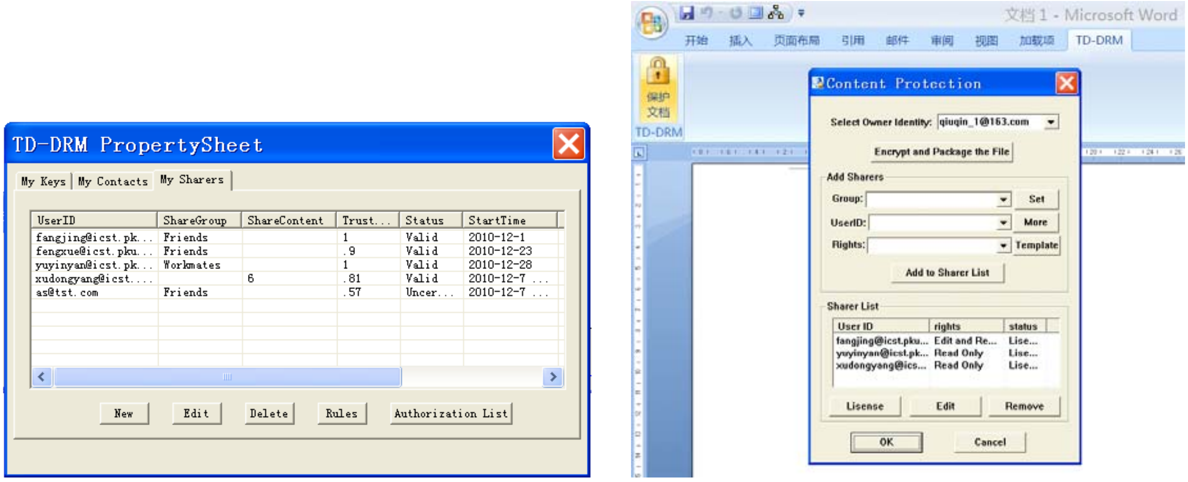
In the underlying trust model, we consider the contexts of authentication and authorization, and allow system users to set different thresholds for trust decision in different scenarios. The trust propagation procedure combines some existing achievements in opinion combination [15, 18] and eliminates the problem of opinion dependence.

Acknowledgment

This work was supported by Major Scientific and Technological Project of GAPP in China (No. GXTC-CZ-1015004).

TABLE II. Comparison of Related Schemes

	[21, 22, 23]	**[8, 9]**	**[3, 10]**	**Our Scheme**
Usage Scenario	Content retail	Content sharing	Content sharing	Content sharing
Autonomous protection	No	No	Yes	Yes
Supports open sharing	No	No	No	Yes
Authorization mode	Content based	Content based	Content based	Content based & Group based



UserID	ShareGroup	ShareContent	Trust...	Status	StartTime
fangjing@icst.pk...	Friends		1	Valid	2010-12-1
fengxue@icst.pku...	Friends		.9	Valid	2010-12-23
yuyinyan@icst.pk...	Workmates		1	Valid	2010-12-28
xudongyang@icst....		6	.81	Valid	2010-12-7 ...
as@tst.com	Friends		.57	Uncer...	2010-12-7 ...

Select Owner Identity: qiuqin_1@163.com

Group:

UserID:

Rights:

User ID	rights	status
fangjing@icst.pku...	Edit and Re...	Lise...
yuyinyan@icst.pk...	Read Only	Lise...
xudongyang@ics...	Read Only	Lise...

(a) (b)

Figure 5. UI of the prototype system: (a) the desktop manager, and (b) the plug-in

REFERENCES

[1] J. S. Erickson, “Fair Use, DRM, and Trusted Computing,” Communications of the ACM. U.S., vol. 46, no. 4, pp. 34-39, 2003.

[2] W. Ku and C. H. Chi, “Survey on the Technological Aspects of Digital Rights Management, “ in ISC 2004. LNCS, vol. 3225, K. Zhang, and Y. Zheng, Eds. Heidelberg: Springer, 2004, pp. 391-403.

[3] S. Bhatt, R. Sion, and B. Carbunar, “A Personal Mobile DRM Manager for Smartphones,” Computers & Security. U.S. issue 28, pp. 327-340, 2009.

[4] Z. Zhang, Q. Pei, and J. Ma, et al., “Security and Trust in Digital Rights Management: A Survey,” International Journal of Network Security. Taiwan, vol. 9, no.3, pp. 247-263, 2009.

[5] S. P. Marsh, “Formalizing Trust as a Computational Concept,” PhD Thesis, Department of Mathematics and Computer Science, University of Stirling, 1994.

[6] C. R. McInerney and S. Mohr, “Trust and knowledge sharing in organizations: Theory and Practice,” RETHINKING KNOWLEDGE MANAGEMENT, Information Science and Knowledge Management. U.S., vol. 12, pp. 65-86, 2007.

[7] Jaehong Park, Yuan Cheng, and R. Sandhu, “towards a framework for cyber social status based trusted open collaboration,” Proc. 6th International Conference on Collaborative Computing: Networking, Applications and Worksharing (CollaborateCom), 2010, pp. 1-8.

[8] Microsoft IRM: http://office.microsoft.com/en-in/excel-help/information-rights-management-in-the-2007-microsoft-office-system-HA010102918.aspx [retrieved: January, 2011].

[9] Voltage SecureFile: http://www.voltage.com/products/sfclient.htm [retrieved: January, 2011].

[10] X. Feng, Q. Qiu, and Z. Tang, “Copy Protection for Email,” Proc. 12th International Conference on Electronic Commerce, 2010, pp. 177-183.

[11] A.-A. Rahman, “The PGP Trust Model,” The journal of Electronic Commerce, 1997.

[12] F. Almenarez, A. Marin, C. Campo, and G. R. Carlos, “PTM: A Pervasive Trust Management Model for Dynamic Open Environments,” Proc. IFIP International Conference on Pervasive Computing and Communications, Pisa, Italy, 2006, pp. 267-271.

[13] Y. Sun, W. Yu, Z. Han, and K. J. Liu, “Information Theoretic Framework of Trust Modeling and Evaluation for Ad Hoc Networks,” IEEE Journal on Selected Areas in Communications, special issue on security in wireless ad hoc networks, vol 24, no.2, pp. 305-317, 2006.

[14] A.-A. Rahman and S. Halles, “A Distributed Trust Model,” Proc. 1997 workshop on new security paradigms, 1998, pp. 48-60.

[15] A. Josang, “An Algebra for Assessing Trust in Certification Chains,” Proc. Network and Distributed Systems Security Symposium (NDSS), 1999, pp. 1-10.

[16] I. Ray, I. Ray, and S. Chakraborty, “An Interoperable Context-Sensitive Model of Trust,” Journal of Intelligent Information Systems, vol 32(1), pp. 1-12, 2009.

[17] Z. Liu, S. S. Yau, D. Peng, and Y. Yin, “A Flexible Trust Model for Distributed Service Infrastructures,” Proc. 11th IEEE Symposium on Object Oriented Real-Time Distributed Computing, 2008, pp. 108-115.

[18] A. Josang and M. Daniel, “Strategies for Combining Conflicting Dogmatic Beliefs,” Proc. 6th International Conference on Information Fusion, 2003, pp. 1133-1140.

[19] U. Blumenthal, N. C. Hien, and B. Wijnen, “Key Derivation for Network Management Applications,” IEEE Network, pp. 26-29, May/June, 1997.

[20] M. J. Atallah, M. Blanton, N. Fazio, and K. B. Frikken, “Dynamic and Efficient Key Management for Access Hierarchies,” ACM Transactions on Information and System Security (TISSEC), vol. 12, issue 3, Article 18, pp. 1-43, 2009.

[21] Architecture of Windows Media Rights Manager: http://www.microsoft.com/windows/windowsmedia/howto/articles/drmarchitecture.aspx [retrieved: May, 2011].

[22] Adobe Content Server: http://www.adobe.com/products/content-server.html [retrieved: April, 2011].

[23] Microsoft DAS: http://www.microsoft.com/reader/developers/info/das.aspx [retrieved: March, 2012].