A Decentralized Authorization Scheme for DRM in P2P File-sharing Systems

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

***Abstract*—To enable secure content trade between peers with good scalability and efficiency in Peer-to-Peer (P2P) file-sharing systems, we propose a decentralized authorization scheme for Digital Rights Management (DRM). Based on proxy re-encryption mechanism, our scheme deploys authorization functions on semi-trusted authorization proxy peers who re-encrypt the ciphertext of content keys and issue licenses to content user peers. A trusted server ensures that content users are properly charged for authorization. Our scheme has no alteration to the structure of P2P network and can be implemented with a wide range of proxy re-encryption algorithms. Compared with authorization schemes in existing DRM solutions, our scheme relieves the server from intense authorization processing, and achieves reliable decentralized authorization with good scalability and efficiency.**

Keywords-DRM; P2P; decentralized authorization; proxy re-encryption

I. Introduction

Peer-to-Peer (P2P) file sharing is an attractive way for content distribution among users with low cost, high scalability and efficiency; however, a large portion of the P2P distribution is illegal and violates copyright laws [1]. There is an urgent need to integrate Digital Rights Management (DRM) solutions into P2P systems for copyright protection. Furthermore, as personal users are playing more and more active roles in the creation and consumption of online content, we predict that there is huge potential market for personal users to sell original content through platforms that can provide convenient content distribution, secure billing and DRM functions. P2P file-sharing system is a good prototype to develop such platforms if DRM solutions can be integrated efficiently.

DRM refers to technologies that support legal distribution of digital media while protecting appropriate property rights. A DRM system needs to address three major areas [2]: (1) ensuring that the digital asset is packaged in a form that will prevent unauthorized usage, (2) appropriately distributing the digital asset and rights to the end-user, and (3) making sure the end-user is able to render the digital asset consistently with his/her rights.

Most existing DRM systems are employed by medium and large content providers in client-server communication context. The provider sets up a license server to authorize users and issue licenses with which users can decrypt and use protected content [2-4]. The authorization method is highly centralized with all authorization related functions, usually including request verification, user authentication, billing, content key retrieval and encryption, and license generation and issuing, placed on the license server.

For individual and small content providers to sell content through P2P network, existing centralized authorization method for DRM is inapplicable. Firstly, the license server would work for a great deal of providers and users in the open and scalable environment; it may be overloaded when the scale of users grows too large. Besides, authorization is supposed to be dynamic and flexible. With limited investment for a high-performance license server and strong network backbones, the traditional centralized authorization method would cause bottleneck problem. Therefore, we propose a decentralized authorization scheme for DRM to maintain the high efficiency and scalability of P2P network while ensuring copyright security.

Our decentralized authorization scheme is based on proxy re-encryption mechanism. Proxy re-encryption is a cryptosystem with the special property that a proxy, given special information, can efficiently convert a ciphertext for Alice into a ciphertext of the same message for Bob without learning either party's secret key or the contents of the message it re-encrypts [14].

The contributions of this paper include: (1) we propose a DRM model for P2P network, which provides common service infrastructure for individual and small content providers who cannot construct a specialized DRM system for their own; (2) as the key component of the DRM model, a decentralized authorization scheme based on proxy re-encryption mechanism is proposed. Our scheme has better scalability and efficiency than traditional centralized authorization schemes. Compared with other decentralized authorization schemes, our scheme is reliable and applicable.

II. Related Work

A lot of DRM solutions have been proposed for either traditional client-server context or P2P network. We divide them into two categories according to their authorization methods.

A. Centralized Authorization Schemes

Most existing DRM systems are developed in conventional client-server communication context. The

978-1-4244-8790-5/11/$26.00 ©2011 IEEE

authorization method is highly centralized with all authorization requests processed by the license server [2-4].

To protect copyrights in P2P file-sharing systems, some researchers work on integrating DRM functions into P2P systems, and they more or less inherit the centralized authorization method from client-server based DRM systems. Iwata et al. [5] presented three DRM system models where content package and usage control functions are deployed on the DRM server or content owner terminals while encrypted contents are distributed through P2P file sharing. Chen et al. [7] proposed a copyright protection solution for BitTorrent-like P2P systems. A tracker site is involved in each content transference to respond authorization request, compute re-encryption keys, and provide decryption keys to target users. Chen et al. [8] presented a revised DRM mechanism for BitTorrent-like P2P system. To use the content encrypted by content owners, content users have to get licenses from content owners after some payment through a trusted C2C payment gateway.

In the above centralized authorization schemes, a high-performance server or content owner terminals and strong network backbones are required for supporting substantial simultaneous authorization requests, which increases the investment and poses great challenges to the scalability and efficiency of P2P systems.

B. Decentralized Authorization Schemes

To keep the high efficiency and scalability of P2P systems while protecting copyrights, some DRM solutions have achieved certain innovation on decentralized authorization, but they either suffer from security vulnerabilities or have special demands on usage scenarios.

In the scheme proposed by Lou et al. [1], multiple trusted agents who are set up by the content server are in charge of authenticating customer peers, distributing digital content to paid customers and preventing unpaid peers from downloading usable content via content poisoning. A customer can download usable unencrypted content from agents and other customers with a valid token signed by an agent. The problem with Lou's scheme is that copyright protection is limited in content distribution within P2P network. Without encryption to content and usage control mechanisms on clients, the scheme is not capable of preventing a customer from purchasing the digital content and then distributing it out of P2P systems.

Sung et al. [6] proposed DRM enabled P2P architecture with no server. The license issue-able right of a peer can be transferred or copied to other peers after some payment, resulting in that any peer in P2P network can issue a license. However, without any controlling mechanisms, such scheme is unreliable. An evil license-issuing peer may issue licenses or copy license issue-able right to other peers without asking for proper payment, making rights management out of control.

Based on broadcast encryption, Zhang et al. [9] proposed a scheme where License Server organizes peers with the same rights to certain contents into an authorized domain; peers in a domain share content keys. The scheme causes License Server not have to transmit the content key for every authorization request; however, each time new customers join a domain, License Server has to notify all domain members of the membership change, and broadcast the updated ciphertext for domain members to re-compute content keys. Too frequent domain joining will degrade the system into client-server networking. Based on Zhang's scheme [9], Liu et al. [10] run a modified scheme with a decentralized key management mechanism by deploying license agencies in each domain, who are peers, as distributed license servers to compute key parameters collaboratively or separately. Both Zhang's scheme [9] and Liu's scheme [10] are only advantageous in scenarios with good-structured domain organization.

III. Our Scheme

Our decentralized authorization scheme is based on proxy re-encryption mechanism. We describe it through a DRM system for secure content trade in P2P network.

A. System Model

The model of our DRM system for P2P network is shown in Fig. 1. It provides common infrastructure for peer nodes to trade content securely.

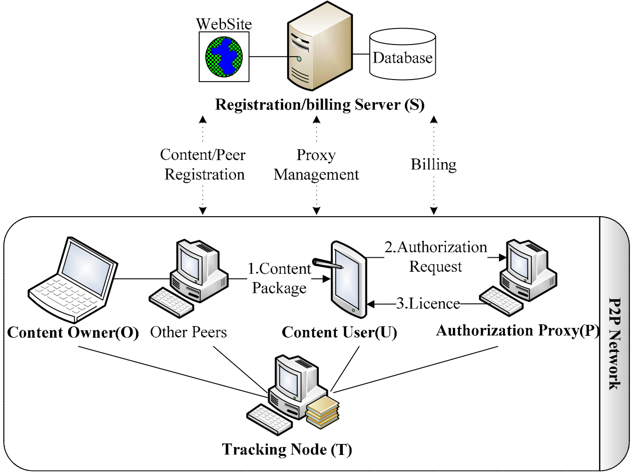

Figure 1. The proposed system model.

The system participants include a trusted registration/billing server, content owners, content users, authorization proxies and at least a tracking node.

The trusted server hosts a website and database. It is in charge of peer registration, content registration, proxy management, and billing transactions. It owns a public/private key pair generated by an external certificated authority (CA) and publishes its public key in the system.

Each participating peer in P2P network can play one or more roles of a content owner, a content user, and an authorization proxy. It installs secure DRM client software and has a public/private key pair which can be generated by the DRM client software, the server, or the external CA.

Tracking nodes exist in P2P network to track requested resources. In our system, they are also responsible for tracking available authorization proxies. They may be super nodes, the P2P server, or peer nodes, depending on the topology of P2P network. We do not change the original structure of P2P network.

The general process of our scheme is as follows: first, the server generates re-encryption keys for registering content users, issues qualification certificates to authorization proxies, and packages each encrypted content with the corresponding ciphertext of content encryption key uploaded by content owners. Content packages can be distributed through P2P file sharing. To use encrypted content, a content user sends an authorization request to an authorization proxy, which contains the ciphertexts of the content encryption key and the content user's re-encryption key. After having the content user charged via the server, the authorization proxy re-encrypts the ciphertext of content encryption key with the content user's re-encryption key, and finally issues him a license containing the re-encrypted content encryption key; the proxy's qualification certificate is also sent for the content user to do verification.

B. Proposed Protocols

Our scheme consists of five phases, as described in the five protocols below. The protocols can be implemented with a wide range of proxy re-encryption algorithms, including the atomic proxy encryption by Blaze et al. [11], the unidirectional proxy encryption by Ivan et al. [12], and more recent algorithms by Ateniese et al. in [13] and [14].

The notations in Table I are used throughout this paper.

TABLE I. NOTATIONS

Notation	Description
S, O, U, P, T	The server, a content owner, a content user, an authorization proxy and a tracking node respectively
M , C	Plaintext/ciphertext of content
CEK	Content encryption key
cp	Content package
CID	Content identifier
R	Rights information of content
ur	Usage rights requested by a content user
ID_X	Identifier of node X
X_addr	Address of node X
pk_X , sk_X	Public/private key of node X
$r_{A\to B}$	Re-encryption key used by a proxy to transform a message encrypted under pk_A to one that can be decrypted with sk_B
RKGen(A,B)	$r_{A\to B}$ generator with the input of sk_A and sk_B or pk_B .
$REnc(r_{A\to B},\bullet)$	Proxy re-encryption with re-encryption key $r_{A\to B}$
QC(P)	Qualification certificate of authorization proxy P
aKey	Secret key generated by the server and shared by authorization proxies

Enc(K,•)
Dec(K,•)
	Secret key encryption/decryption of a message with symmetric key K

$PEnc(pk_X,\bullet)$ $PDec(sk_X,\bullet)$	Public key encryption with public key pk_X ; Public key decryption with private key sk_X
$HSign_X$	X's signature over the hash digest of information

1) Peer registration and re-encryption key generation

First of all, participating peers in P2P network have to register to the server S; each of the peers can play one or more roles of a content owner, a content user and an authorization proxy. S computes a re-encryption key for each registering content user.

As shown in Fig. 2, when peer U registers to be a content user, S generates $r_{S\to U}$ with sk_S and pk_U or sk_U (depending on the adopted proxy re-encryption algorithm), encrypts $r_{S\to U}$ with pre-generated secret key aKey as $\alpha = Enc(aKey, r_{S\to U})$, and sends $\overline{\alpha} = \{ID_U, \alpha, HSign_S\}$ to U. $\overline{\alpha}$ will be used for U to request authorization.

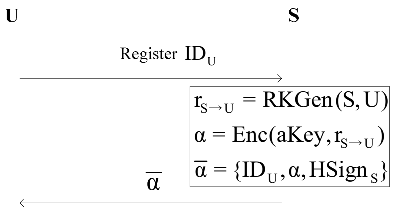
U
S
Register ID_U
$r_{S\to U} = RKGen(S,U)$
$\alpha = Enc(aKey, r_{S\to U})$
$\overline{\alpha} = \{ID_U, \alpha, HSign_S\}$
$\overline{\alpha}$

Figure 2. Peer registration and re-encryption key generation.

2) Authorization proxy initialization

When S appoints peer P to be an authorization proxy (P is semi-trusted; it holds no secret and follows the procedure), S generates qualification certificate QC(P) , encrypts pre-generated aKey with pk_P as $\delta = PEnc(pk_P, aKey)$, and sends QC(P) and δ to P, as shown in Fig. 3. Meanwhile, P_addr is registered to the related tracking node T.

QC(P) contains S's signature on proxy information such as commission rate and valid period; it will be used by content users to verify P's legality as an authorization proxy. δ is used for P to get aKey securely.

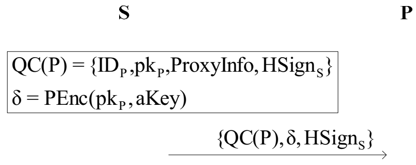
S
P
$QC(P) = \{ID_P, pk_P, ProxyInfo, HSign_S\}$
$\delta = PEnc(pk_P, aKey)$
$\{QC(P), \delta, HSign_S\}$

Figure 3. Authorization proxy initialization.

3) Content registration and package

Content owner O uploads content ciphertext C, rights information R, and $\beta = PEnc(pk_S, CEK)$ to S for content registration, as in Fig. 4. S can use CEK recovered from β to decrypt C for sensitive (sexual, terrorized, etc.) content detection if necessary. Next, S assigns a unique content identifier CID to the content, generates $\overline{\beta} = \{CID, \beta, HSign_S\}$, and packages CID, $\overline{\beta}$ and other content information as content package cp . Containing CID and the corresponding ciphertext of CEK, $\overline{\beta}$ will be used for content users to request authorization on CID.

Once cp gets into P2P network from the server by http downloading or other ways, it can be distributed among peers through P2P file sharing.

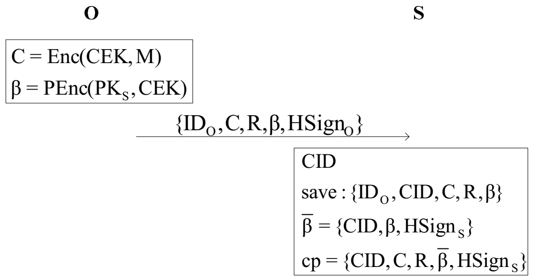
O
S
$C = Enc(CEK, M)$
$\beta = PEnc(PK_S, CEK)$
$\{ID_O, C, R, \beta, HSign_O\}$
CID
$save: \{ID_O, CID, C, R, \beta\}$
$\overline{\beta} = \{CID, \beta, HSign_S\}$
$cp = \{CID, C, R, \overline{\beta}, HSign_S\}$

Figure 4. Content registration and package.

4) User authorization

After getting content package cp, content user U requests a proxy for authorization with usage rights ur on the protected content CID in cp. The process is as follows (as shown in Fig 5):

a) Authorization request

First, U queries T authorization proxy addresses. After getting addresses of available authorization proxies from T, U selects P's address and sends P an authorization request. The authorization request contains $\overline{\alpha}$ which was sent from S to U in peer registration phase, and $\overline{\beta}$ which was collected from cp.

b) Authorization processing

P verifies S's signature in $\overline{\alpha}$ and $\overline{\beta}$, then generates transaction information from the authorization request for S to do billing operations.

After getting billing success message from S, which contains anti-replay information such as the digest of the transaction information, P's DRM client is triggered to do the following operations: first, P collects α and β from $\overline{\alpha}$ and $\overline{\beta}$ respectively; then, P recovers aKey from δ which was sent from S to P in the phase of authorization proxy initialization; next, P decrypts α with aKey for $r_{S\to U}$, and computes $\gamma = REnc(r_{S\to U}, \beta)$; finally, S generates a license with γ, and sends the license, together with QC(P), to U.

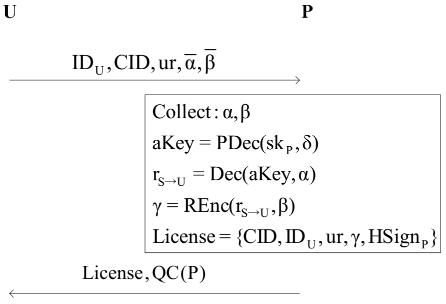
U
P
$ID_U, CID, ur, \overline{\alpha}, \overline{\beta}$
$Collect: \alpha, \beta$
$aKey = PDec(sk_P, \delta)$
$r_{S\to U} = Dec(aKey, \alpha)$
$\gamma = REnc(r_{S\to U}, \beta)$
$License = \{CID, ID_U, ur, \gamma, HSign_P\}$
$License, QC(P)$

Figure 5. User authorization.

5) Content usage

On receiving the license and $QC(P)$, U's DRM client first verifies $HSign_S$ in $QC(P)$, then verifies $HSign_P$ in the license with pk_P collected from $QC(P)$:

If either the license or $QC(P)$ is invalid, U rejects using the license and reports to S.

If both the license and $QC(P)$ are valid, U fetches γ from the license and recovers CEK. Finally, U can decrypt C with CEK and use M according to ur in the license.

$$CEK = PDec(sk_U, \gamma), \text{ where } \gamma = REnc(r_{S\to U}, \beta)$$
$$M = Dec(CEK, C)$$

IV. Analysis

A. Server Overheads

We relieve the server from intense and frequent authorization processing by having authorization proxies do re-encryption and license issuing operations with α, β, and $QC(P)$ that are pre-generated by the server in peer registration phase, content registration phase and proxy initialization phase respectively. The server only performs billing operation in user authorization phase, without doing hard work such as content encryption key retrieval and encryption, as well as license generation and issuing.

Although the server is in charge of peer registration and authorization proxy initialization, they are one-time operations for each registering peer and authorization proxy. In the content registration phase, the main task of the server is only content package because the content encryption key generation and content encryption have been completed on content owner terminals. Obviously, a specialized content server can be introduced if necessary.

The server is properly loaded in our scheme.

B. Proxy Overheads

Proxy re-encryption imposes overheads on proxies to different extent, depending on the selected re-encryption algorithm. Some algorithms just involve a light multiply operation [11]; some involve a heavy bilinear pairing and other operations [13]. However, even when running a complex algorithm with a bilinear pairing, the cost of re-encryption is acceptable for a client machine according to Ateniese's experimental results [13].

Besides, as authorization proxies hold no secrets, we do not require the proxies to be completely trusted. Therefore, the total authorization overhead can be amortized to a large number of proxies. How to pick up authorization proxies is related to the business policy in actual applications.

C. System Security

The security of our scheme relies on the trusted server and secure DRM client software, which are realizable and practical currently. The trusted server enforces that a user must pay for requested rights to get the corresponding license bound to his private key from an authorization proxy. Secure DRM client software ensures only legal operations on user terminals and authorization proxy terminals without giving out any secret information.

In particular, our protocol ensures the security of authorization in two aspects. Firstly, only a qualified authorization proxy with a valid qualification certificate can issue licenses that can be accepted by uncompromised content users; secondly, only after getting billing success message from the server, can the DRM client of an authorization proxy do re-encryption operation and issue licenses in the way described in the protocol.

Attackers may recover sk_S from $r_{S \rightarrow U}$ and sk_U if the adopted proxy re-encryption algorithm is not collusion safe [13]; however, our scheme can prevent such attack. Firstly, because $r_{S \rightarrow U}$ that S issues to U is encrypted with aKey and can only be decrypted in the DRM client of authorization proxies under constraints, attackers cannot get its plaintext for the attack; secondly, with secure DRM client software or other existing techniques, sk_U can also be kept secret without leakage. Of course, it achieves better security to use a collusion-safe proxy re-encryption algorithm, such as those proposed in [13, 14], so that attackers cannot get sk_S even with both the plaintexts of $r_{S \rightarrow U}$ and sk_U.

In the case that a compromised authorization proxy is discovered, the server publishes the revocation of his qualification certificate.

V. Comparisons and Conclusions

In this paper, we propose a decentralized authorization scheme for DRM in P2P network. By adopting proxy re-encryption mechanism, we deploy authorization functions on authorization proxies who are semi-trusted peer nodes in P2P network; for security, a trusted server manages registration and billing.

A comparison of the main authorization operations in typical centralized authorization schemes and our scheme is shown in Table II. Our scheme relieves the server from intense authorization operations, balances system overheads, and lifts authorization efficiency by enabling the concomitant processing of authorization requests on different proxies.

Among existing decentralized authorization schemes, Lou's scheme [1] only ensures legal distribution of plain content within P2P network by employing trusted agents supervising distribution activities; Sung's scheme [6] is unreliable in enabling license issuing on any peer terminal without any controlling mechanisms; Zhang's scheme [9] and Liu's scheme [10] are only advantageous in scenarios with good-structured domain organization. By comparison, our decentralized scheme reaches reliable and persistent copyright protection; it is applicable for secure content trade with good scalability and efficiency in P2P network.

Acknowledgment

We thank Xue Feng, Xiaoyan Lin and Dongyang Xu for their valuable insights. Our work is supported by the National High Technology Research and Development Program of China (No.2009AA01Z408) and Beijing Natural Science Foundation (No.4082014).

References

[1] X.S. Lou, K. Hwang, and R.F. Zhou, "Integrated Copyright Protection in Peer-to-Peer Networks," 27th International Conference on Distributed Computing Systems Workshops (ICDCSW 07), Jun. 2007, pp. 28-28.

[2] W.J. Zeng, H. Yu, and C.Y. Lin, Multimedia Security Technologies for Digital Rights Management. Academic Press, USA, 2006.

[3] W. Rosenblatt, W. Trippe, and S. Mooney, Digital Rights Management: Business and Technology. M&T Books, New York, 2002.

[4] Microsoft Windows Media Rights Manager.
http://www.microsoft.com/windows/windowsmedia/howto/articles/drmarchitecture.aspx

[5] T. Iwata, T. Abe, K. Ueda, and H. Sunaga, "A DRM system suitable for P2P content delivery and the study on its implementation," The 9th Asia-Pacific Conference on Communications (APCC 2003), vol. 2, Sept. 2003, pp. 806-811.)

[6] J.Y. Sung, J.Y. Jeong, and K.S. Yoon, "DRM Enabled P2P Architecture," The 8th International Conference on Advanced Communication Technology (ICACT 2006), vol. 1, Feb. 2006, pp. 487-490.

[7] S.Q. Chen, and X.W. Zhang, "Digital Rights Protection in BitTorrent-like P2P Systems," United States Patent Application, US2009/0210697, Aug. 2009.

[8] Y.Y. Chen, J.K. Jan, Y.Y. Chi, and M.L. Tsai, "A Feasible DRM Mechanism for BT-Like P2P System," Information Engineering and Electronic Commerce (IEEC 09), May 2009, pp. 323-327.

[9] Y. Zhang, C. Yuan, and Y.Z. Zhong, "Implementing DRM over Peer-to-Peer Networks with Broadcast Encryption," 8th Pacific Rim Conference on Multimedia (PCM 2007), LNCS, vol. 4810, Springer Press, Dec. 2007, pp. 236-245.

[10] L. Liu, and C. Yuan, "Broadcast Encryption-Based P2P DRM without Central License Server," 10th Pacific Rim Conference on Multimedia (PCM 2009), LNCS, vol. 5879, Springer Press, Dec. 2009, pp. 451-458.

[11] M. Blaze, and M. Strauss, "Atomic Proxy Cryptography," International Conference on the Theory and Application of Cryptographic Techniques (Eurocrypt 98), May 1998

[12] A. Ivan, and Y. Dodis, "Proxy Cryptography Revisited," Proc. 10th Annual Network and Distributed System Security Symposium (NDSS 2003), Feb. 2003.

[13] G. Ateniese, K. Fu, M. Green, and S. Hohenberger, "Improved proxy re-encryption schemes with applications to secure distributed storage," ACM Transactions on Information and System Security (TISSEC), vol. 9, No. 1, Feb. 2006, pp. 1-30.

[14] G. Ateniese, K. Benson, and S. Hohenberger, "Key-private proxy re-encryption," RSA Conference 2009, Cryptographers' Track (CT-RSA '09), LNCS, vol. 5473, Springer Press, Apr. 2009, pp. 279-294, doi: 10.1007/978-3-642-00862-7_19.

TABLE II. Comparison of Main Authorization Operations

	The Central Server	**Authorization Proxies**
Centralized Schemes	1. billing 2. key retrieval/decryption 3. key encryption 4. license generation	
Our Scheme	1. billing	1. key re-encryption 2. license generation