Available online at www.sciencedirect.com
ScienceDirect

ELSEVIER

December 2013, 20(6): 88–95
www.sciencedirect.com/science/journal/10058885

The Journal of China Universities of Posts and Telecommunications

http://jcupt.xsw.bupt.cn

Secure and privacy-preserving DRM scheme using homomorphic encryption in cloud computing

HUANG Qin-long[1,2,3] (✉), MA Zhao-feng[1,2,3], YANG Yi-xian[1,2], FU Jing-yi[1,2,3], NIU Xin-xin[1,2]

1. Information Security Center, Beijing University of Posts and Telecommunications, Beijing 100876, China
2. National Engineering Laboratory for Disaster Backup and Recovery, Beijing University of Posts and Telecommunications, Beijing 100876, China
3. Beijing National Security Science and Technology Co. Ltd, Beijing 100086, China

Abstract

Cloud computing provides a convenient way of content trading and sharing. In this paper, we propose a secure and privacy-preserving digital rights management (DRM) scheme using homomorphic encryption in cloud computing. We present an efficient digital rights management framework in cloud computing, which allows content provider to outsource encrypted contents to centralized content server and allows user to consume contents with the license issued by license server. Further, we provide a secure content key distribution scheme based on additive homomorphic probabilistic public key encryption and proxy re-encryption. The provided scheme prevents malicious employees of license server from issuing the license to unauthorized user. In addition, we achieve privacy preserving by allowing users to stay anonymous towards the key server and service provider. The analysis and comparison results indicate that the proposed scheme has high efficiency and security.

Keywords digital rights management, homomorphic encryption, proxy re-encryption, privacy preserving, cloud computing

1 Introduction

The cloud computing is the fundamental change happening in the field of information technology which provides services, computation, and storage from a remote and centralized facility or contractor [1]. The cloud computing is clearly one of today's most exciting technologies, at least in part due to its cost-efficiency and flexibility. One of the most important characteristics of cloud computing is its pay-as-you-go manner.

In cloud computing, digital contents can be easily and ubiquitously accessed. However, several security issues in the cloud are impeding the vision of cloud computing as a new IT procurement model, content security in the cloud is one of them. The trading and sharing of digital contents are very convenient with the rapid development and growth of the Internet and cloud computing. However, a lot of digital contents have been pirated and illegally distributed [2]. DRM is the important technology of taking a series of measures to protect the digital contents from being abused and make sure that digital contents are fair to use [3–4].

Nowadays many DRM schemes have been proposed in cloud computing. Proxy re-encryption (PRE) is widely used to protect the content in semi-trusted cloud environment. In the proxy re-encryption, a semi-trusted proxy computes a function that converts a ciphertext under Alice's public key into another ciphertext that can be opened by Bob's secret key without seeing the underlying plaintext. Thus encrypted content or content key can be re-encrypted to allow the user to decrypt the encrypted content or content key, while DRM server cannot be able to get the plain content or content key. However, malicious employees of DRM server may re-encrypt content or content key to any users without letting content provider know.

Therefore, we propose a secure and privacy-preserving DRM scheme using homomorphic encryption in cloud computing. Our contributions are as follows:

1) We present an efficient DRM framework in cloud

Received date: 01-05-2013
Corresponding author: HUANG Qin-long, E-mail: longsec@bupt.edu.cn
DOI: 10.1016/S1005-8885(13)60113-2

computing, which allows content provider to outsource encrypted contents to centralized content server and allows user to consume contents with the license issued by license server.

2) We provide a secure content key distribution scheme based on additive homomorphic probabilistic public key encryption and proxy re-encryption. The license server only can re-encrypt the content main key which is part of the content encryption key, while the key server computes the encrypted content encryption key which is further distributed to the user in the license based on additive homomorphic probabilistic public key encryption. Thus the provided scheme prevents the malicious employees of license server from issuing license to unauthorized user.

3) We achieve privacy preserving by allowing users to stay anonymous towards the key server and service provider.

The remainder of this paper is organized as follows: related work is covered in Sect. 2 and the preliminaries in Sect. 3. We come up with requirements for DRM scheme in cloud computing in Sect. 4. We present a secure and privacy-preserving DRM scheme in Sect. 5. We analyze the performance in Sect. 6. Finally we conclude in Sect. 7.

2 Related work

In the academia, several DRM schemes have been proposed in cloud computing. Wang et al. proposed a cloud-based DRM (CS-DRM) scheme for the mobile Internet [5]. The characteristics of cloud computing enable CS-DRM to bring benefits for content providers, and well satisfy the performance requirements with low cost when the number of users increases significantly. However, the proposed scheme requires a trusted cloud computing environment to protect contents and preserve users' privacy.

In order to support the requirements for semi-trusted environments, numbers of DRM schemes have also been proposed based on proxy re-encryption [6–7]. Lee et al. proposed a secure mutual-profitable DRM interoperability scheme [6]. The proposed scheme uses a designated proxy re-encryption scheme and minimizes disclosure of the security properties of DRM technology providers and content providers while preserving their profits. He et al. proposed a novel DRM infrastructure which is based on a non-transferable re-encryption scheme [7]. In the proposed infrastructure, DRM technology providers and content providers are required to cooperate to make a purchased digital content for a specific device accessible by other different devices. Although these schemes can protect contents well and prevent malicious employees of DRM server from issuing licenses without letting content provider know, they do not consider user privacy.

Recently, a number of cloud computing security schemes based on homomorphic encryption have been proposed. Corena et al. proposed a novel architecture based on additive homomorphic encryption and secret sharing schemes to store information securely while still allowing fast aggregation queries at an outsourced untrusted cloud server [8]. Samanthula et al. proposed an efficient and secure data sharing framework in the cloud using homomorphic encryption and proxy re-encryption schemes [9]. In this framework, upon a data request from the user, the cloud computes the re-encrypted record using the re-encryption key and performs a homomorphic addition to generate a set of attributes, and sends the results to the user for decryption.

In order to preserve user privacy in cloud computing, Petrlic et al. proposed a privacy-preserving DRM concept for cloud computing [10]. The proposed scheme employs a secret sharing scheme based on homomorphic encryption and combines it with a re-encryption scheme to achieve privacy protection. Petrlic also presented a privacy preserving cloud DRM scheme that allows users to stay anonymous and that prevents any party from building user profiles [11]. The proposed scheme extends the proxy re-encryption scheme by Ateniese et al. [12] to achieve indistinguishability of first-level ciphertext under the condition that the same second-level ciphertext is re-encrypted for the same party more than once. However, these schemes need to re-encrypt the content every time when the user consumes the content.

In contrast to the scheme proposed by Petrlic [11], which is based on proxy re-encryption, we propose a novel DRM scheme using homomorphic encryption and proxy re-encryption in cloud computing, which protects the content and privacy without re-encrypting the content.

3 Preliminaries

Our scheme relies upon homomorphic encryption [13–14] and proxy re-encryption [12–15]. We will briefly present some properties related to these techniques.

3.1 General DRM system

The general DRM system involves the content provider, content server, license server and user [16–17]. Fig. 1 shows the framework of general DRM system.

1) Content provider: content provider holds the digital content and wants to protect the content from illegal consumption. Thus, content provider encrypts the content and provides encrypted content to content server, and provides encryption key and usage rules to the license server.

2) Content server: content server keeps the encrypted content over the storage server and provides the encrypted content to user.

3) License server: license server generates the license for the user. The license involves the encryption key and usage rules.

4) User: user gets the encrypted content from content server and acquires the license from license server. The user decrypts the content with encryption key and consumes the content according to the usage rules in the license.

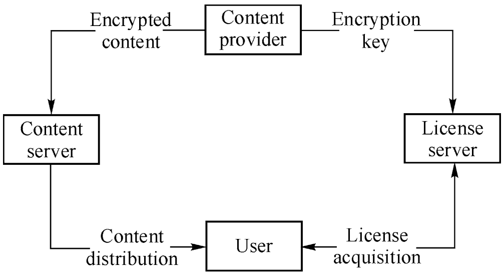

Fig. 1 Framework of general DRM system

3.2 Additive homomorphic probabilistic public key encryption

Homomorphic encryption allows performers to compute correct operations over encrypted values without being aware of their content. This flexibility resolves security issues in a variety of applications that delegate sensitive processing to semi-trusted third parties [13].

The additive homomorphic probabilistic public key encryption (AHPE) [18] involves an encryption function E and a decryption function D. The AHPE exhibits the following properties:

$E(P, m+n) = E(P, m) +_{\mathrm{H}} E(P, n) \bmod N^2$

where m and n are plaintext messages, $+_{\mathrm{H}}$ is the additive homomorphic operation and N is the group size of the encryption scheme.

The encryption function is semantically secure, thus an attacker cannot get any additional information about the plaintext from a given set of ciphertexts.

3.3 Proxy re-encryption

Matt et al. proposed the concept of proxy re-encryption in 1998 [15]. The PRE scheme is a tuple of polynomial time algorithm.

1) Key generation: $G()$. This algorithm generates a public key P and a private key S randomly.

2) Re-encryption key generation: $Z(S_a, P_b)$. Given user a's private key S_a, and user b's public key P_b, this algorithm outputs a key $R_{a\text{->}b}$ that allows re-encrypting ciphertexts intended to a into ciphertexts encrypted for b.

3) Encryption: $E(P, m)$. Given a receiver's public key P and a plaintext m, this algorithm outputs a ciphertext that can be re-encrypted using the appropriate re-encryption key.

4) Re-encryption: $F(R_{a\text{->}b}, c)$. This algorithm takes as input a re-encryption key $R_{a\text{->}b}$ and a ciphertext c encrypted under user a's public key. The output is a ciphertext c' re-encrypted for user b.

5) Decryption: $D(S,c)$. Given a private key S, a ciphertext c, this algorithm returns a plaintext m.

Moreover, for any message and any couple of private/public key pair (S_a, P_a), (S_b, P_b), these algorithms should satisfy the following conditions of correctness:

$D(S_a, E(P_a, m)) = m$

$D(S_b, F(Z(S_a, P_b), E(P_a, m))) = m$

4 Requirements

1) Efficient

In cloud computing the user expects to access the content via multiple devices anytime anywhere, and also asks for flexible usage model. Therefore, the DRM scheme in cloud computing should support efficient license models, and have low computational complexity to support massive users.

2) Security

The content provider expects that digital content must be encrypted and authorized user must not be able to extract and store the decrypted content, and also content confidentiality against unauthorized users must be achieved.

Meanwhile, service provider and license server in the cloud must not be able to get the plain content and content key, malicious employees of license server must not be able to issue license to unauthorized user.

3) Privacy preserving

Privacy preserving is the protection of personal information. The user should stay anonymous towards the key server that generates user's public/private key, and service provider that handles user's content purchase and license acquisition request.

5 Proposed scheme

In this paper, we propose a secure and privacy-preserving DRM scheme based on homomorphic encryption and PRE in cloud computing. To protect the contents stored in the cloud, we present an efficient DRM framework, which allows content provider to outsource encrypted contents to centralized content server. To address the key management challenges, we provide a secure key distribution scheme based on AHPE and PRE. In addition, we allow the users to stay anonymous towards the key server and service provider. The notations are shown in Table 1.

Table 1 Notations in proposed scheme

Notation	Description
CP	Content provider
U	User
SP	Service provider
LS	License server
P	Public key
S	Secret key
R	Re-encryption key
C_{ID}	Content identity
K_{CM}	Content main key
K_{A}	Assistant key
K_{CE}	Content encryption key
M	Plain content data
C	Encrypted content data
W_{U}	User rights
W_{E}	Rights expression
T	Timestamp
$I()$	Signature algorithm
Q_{LS}	License acquisition request signature
Q_{KS}	Key acquisition request signature
Q_{L}	License signature

The framework of proposed scheme consists of the following four phases, as shown in Fig. 2.

1) Key generation: key server generates the public/private key pair for content provider and user, and content provider generates the re-encryption key for authorized user and sends it to license server. Further, key server randomly generates the assistant key and distributes it to content provider. Moreover, content provider generates the random content main key.

2) Content encryption: content provider computes content encryption key by adding content main key and assistant key. Then content provider encrypts the content with content encryption key, and outsources the encrypted content to centralized content server.

3) License acquisition: service provider sends user's license acquisition request to license server. And the license server re-encrypts the content main key which is part of the content encryption key. Then key server computes the encrypted content encryption key with content main key and assistant key based on additive homomorphic probabilistic public key encryption. The license server then generates the license and distributes the license to user through service provider.

4) Content consumption: user checks the license and decrypts the content encryption key with the license before consuming the content. Then the user decrypts the content with content encryption key, and consumes the content according to the usage rules in the license.

With the proposed scheme, content is outsourced to centralized content server in cloud computing only once and thus avoids expensive content re-encryption tasks. We will present detailed descriptions of above phases.

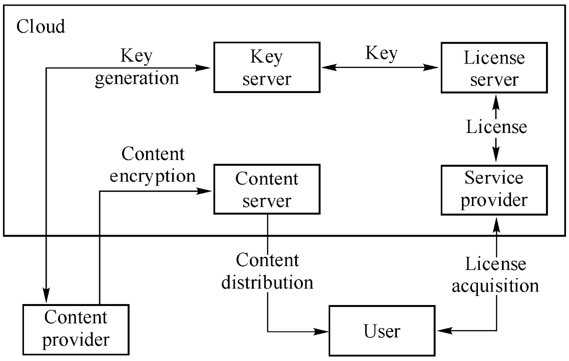

Fig. 2 Framework of proposed scheme

5.1 Key generation

In the key generation phase, content provider generates the re-encryption key for authorized user and sends it to license server. Then the key server generates the assistant key and distributes it to content provider. Moreover, content provider also generates the content main key. The sequence diagram of key generation is shown in Fig. 3.

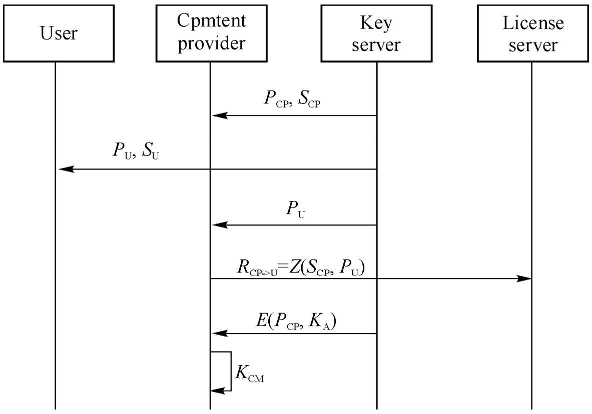

Fig. 3 Sequence diagram of key generation

Step 1 The key server creates a public/private key pair (P_{CP}, S_{CP}) for content provider and sends it to content provider in a secure channel.

Step 2 The key server creates a public/private key pair (P_U, S_U) for anonymous user and sends it to anonymous user in a secure channel. The key server cannot be able to get the user's personal information.

Step 3 Then content provider generates re-encryption key $R_{CP->U}$ for the user using S_{CP} and P_U, and sends it to license server in a secure channel.

$R_{CP->U} = Z(S_{CP}, P_U)$

Step 4 Then key server randomly generates the K_A, and encrypts it with content provider's public key P_{CP}, and then sends the $E(P_{CP}, K_A)$ to content provider. Further, the content provider generates the K_{CM} randomly.

5.2 Content encryption

In the content encryption phase, content provider encrypts the content with content encryption key, and then outsources the encrypted content to centralized content server. The sequence diagram of content encryption is shown in Fig. 4.

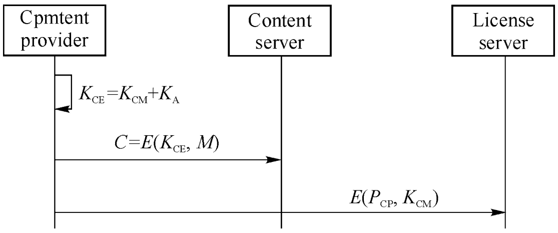

Fig. 4 Sequence diagram of content encryption

Step 1 The content provider decrypts the K_A with the private key SK_{CP}, and then computes K_{CE} by adding K_{CM} and K_A.

$K_{CE} = K_{CM} + K_A$

Step 2 Then content provider encrypts content with K_{CE} and outsources the encrypted content to content server.

$C = E(K_{CE}, M)$

Step 3 Then content provider encrypts the K_{CM} with the public key and sends the $E(P_{CP}, K_{CM})$ to license server.

5.3 License acquisition

In the license acquisition phase, service provider sends user's license acquisition request to license server. The key server computes the encrypted content encryption key based on additive homomorphic probabilistic public key. The license server then generates the license and distributes the license to user through service provider. The sequence diagram of license acquisition is shown in Fig. 5.

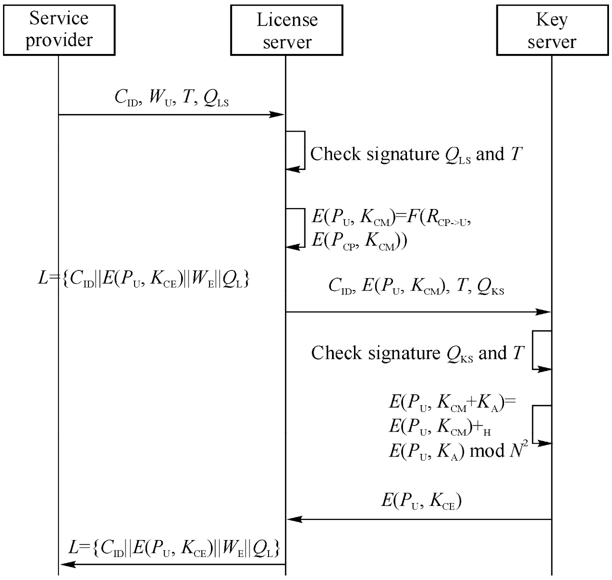
Service provider
License server
Key server
C_{ID}, W_U, T, Q_{LS}
Check signature Q_{LS} and T
$E(P_U, K_{CM})=F(R_{CP->U}$, $E(P_{CP}, K_{CM}))$
$L=\{C_{ID}||E(P_U, K_{CE})||W_E||Q_L\}$
C_{ID}, $E(P_U, K_{CM})$, T, Q_{KS}
Check signature Q_{KS} and T
$E(P_U, K_{CM}+K_A)=$ $E(P_U, K_{CM})+_H$ $E(P_U, K_A)$ mod N^2
$E(P_U, K_{CE})$
$L=\{C_{ID}||E(P_U, K_{CE})||W_E||Q_L\}$

Fig. 5 Sequence diagram of license acquisition

Step 1 The user anonymously purchases interesting content in the service provider, and service provider sends license acquisition request to license server. The license acquisition request includes C_{ID}, W_U, T and signature $Q_{LS} = I(S_{SP}, C_{ID} || W_U || T)$.

Step 2 Then license server checks the signature Q_{LS} and T, and then re-encrypts the $E(P_{CP}, K_{CM})$ with re-encryption key $R_{CP->U}$.

$E(P_U, K_{CM}) = F(R_{CP->U}, E(P_{CP}, K_{CM}))$

Step 3 The license server sends the key acquisition request to key server. The key acquisition request includes C_{ID}, $E(P_U, K_{CM})$, T and $Q_{KS} = I(S_{LS}, C_{ID} || E(P_U, K_{CM}) || T)$.

Step 4 The key server checks the signature Q_{KS} and T, and encrypts the K_{A} with user's public key, and computes the encrypted ($K_{\mathrm{CM}} + K_{\mathrm{A}}$).

$E(P_{\mathrm{U}}, K_{\mathrm{CM}} + K_{\mathrm{A}}) = E(P_{\mathrm{U}}, K_{\mathrm{CM}}) +_{\mathrm{H}} E(P_{\mathrm{U}}, K_{\mathrm{A}}) \bmod N^2$

where $+_{\mathrm{H}}$ is the additive homomorphic property and N is the group size of the encryption scheme. Since $K_{\mathrm{CE}} = K_{\mathrm{CM}} + K_{\mathrm{A}}$, the key server returns $E(P_{\mathrm{U}}, K_{\mathrm{CE}})$ to license server.

Step 5 The license server generates the rights expression W_{E} from the W_{U} according the rights expression language, and then generates the license L which includes content identity C_{ID}, encrypted K_{CE}, usage rights W_{E} and signature

$Q_{\mathrm{L}} = I(S_{\mathrm{LS}}, C_{\mathrm{ID}} \| E(P_{\mathrm{U}}, K_{\mathrm{CE}}) \| W_{\mathrm{E}})$

$L = \{C_{\mathrm{ID}} \| E(P_{\mathrm{U}}, K_{\mathrm{CE}}) \| W_{\mathrm{E}} \| Q_{\mathrm{L}}\}$

Step 6 Then license server sends the L to user through service provider.

5.4 Content consumption

In the content consumption phase, user checks the license, decrypts the content with content encryption key and consumes the content according to usage rules in the license.

Step 1 The user first checks the license when consuming the content, and then decrypts the K_{CE} with the license.

$K_{\mathrm{CE}} = D(S_{\mathrm{U}}, E(P_{\mathrm{U}}, K_{\mathrm{CE}}))$

Step 2 The user then decrypts the content with the K_{CE}. Since the DRM client in user's device has the tamper resistant security module to prevent malicious user from attacking the client, so the decryption process is under the DRM client control. Then the user consumes the content according to the W_{E} in the license.

$M = D(K_{\mathrm{CE}}, C)$

6 Performance analysis

6.1 Correctness

Based on proxy re-encryption scheme, the license server can re-encrypt the encrypted K_{CM} which is provided by content provider for authorized user.

$E(P_{\mathrm{U}}, K_{\mathrm{CM}}) = F(R_{\mathrm{CP\text{-}>U}}, E(P_{\mathrm{CP}}, K_{\mathrm{CM}}))$

Based on additive homomorphic probabilistic public key encryption scheme, the key server computes the encrypted ($K_{\mathrm{CM}} + K_{\mathrm{A}}$).

$E(P_{\mathrm{U}}, K_{\mathrm{CM}} + K_{\mathrm{A}}) = E(P_{\mathrm{U}}, K_{\mathrm{CM}}) +_{\mathrm{H}} E(P_{\mathrm{U}}, K_{\mathrm{A}}) \bmod N^2$

Since $K_{\mathrm{CE}} = K_{\mathrm{CM}} + K_{\mathrm{A}}$, the key server sends $E(P_{\mathrm{U}}, K_{\mathrm{CE}})$ to license server which generates the license for authorized user. Thus the user can decrypt the content, but cannot access the content that he is not authorized to access.

6.2 Security

Proof 1 Only authenticated users with the appropriate license can decrypt the content.

The content provider provides the centralized content server with encrypted content. Thus the user cannot decrypt the content without the content encryption key, and also cannot relay the content to anyone else.

The user only can purchase the content from service provider and acquire the license from the license server. With the license, the user can compute the content encryption key with the private key. Thus only authorized user can decrypt the content with the appropriate license.

Proof 2 The malicious employees of license server cannot issue license to unauthorized user.

In the license acquisition phase, license server only can re-encrypt the K_{CM} which is part of the K_{CE} and generate $E(P_{\mathrm{U}}, K_{\mathrm{CM}})$. Then license server sends $E(P_{\mathrm{U}}, K_{\mathrm{CM}})$ to the key server, and the key server computes the $E(P_{\mathrm{U}}, K_{\mathrm{CE}})$ based on additive homomorphic probabilistic public key encryption. Therefore, malicious employees of license server cannot issue license to unauthorized user without the encrypted K_{CE} generated by key server.

Proof 3 The attackers cannot replay license acquisition request and key acquisition request.

The license acquisition request between service provider and license server is signed by service provider, and the license server checks the signature Q_{LS} and T when receiving the license acquisition request. If the attackers replay license acquisition request and send ($C_{\mathrm{ID}} \| W_{\mathrm{U}} \| T' \| Q_{\mathrm{LS}}$) to license server, the license server checks $T' \neq T$, and refuses the request, thus the attackers cannot replay the license acquisition request.

Moreover, the key acquisition request between license server and key server is signed by license server, and the key server checks the signature Q_{KS} and T when receiving the key acquisition request. If the attackers replay key acquisition request and send ($C_{\mathrm{ID}} \| E(P_{\mathrm{U}}, K_{\mathrm{CM}}) \| T' \| Q_{\mathrm{KS}}$) to key server, the key server checks $T' \neq T$ and refuses the request, thus the attackers cannot replay the key acquisition request.

6.3 Privacy preserving

The user directly communicates with the key server and service provider, and the other parties cannot get any user's personal information.

In the key generation phase, user anonymously registers to the key server which creates the public/private key for the user. In the license acquisition phase, user purchases the content and acquires the license in the service provider without revealing any personal information. Therefore, the user stays anonymous towards key server and service provider, and the user's privacy is preserved.

6.4 Complexity

We use T_a to represent the asymmetric encryption, T_s to represent the symmetric encryption, T_r to represent the proxy re-encryption and T_h to represent the modular addition, the calculation quantity in content preparation phase including content encryption and license acquisition is $7T_a + T_s + 2T_h + T_r$, and the calculation quantity in content consumption phase is $2T_a + T_s$. Therefore, the whole calculation quantity of our scheme is $9T_a + 2T_s + 2T_h + T_r$. Our scheme compared with Petrlic et al.'s scheme [11] is shown in Table 2. Since the computational cost of asymmetric encryption is usually much higher than symmetric encryption and modular addition, our scheme has a lower computational complexity compared to Petrlic et al.'s scheme.

Table 2 Complexity comparison

DRM Scheme	Content preparation	Content consumption	Total
Scheme in Ref. [11]	$5T_a$	$8T_a + T_r$	$13T_a + T_r$
Our scheme	$7T_a + T_s + 2T_h + T_r$	$2T_a + T_s$	$9T_a + 2T_s + 2T_h + T_r$

6.5 Comparison

Our scheme compared with other schemes is shown in Table 3. Our scheme encrypts the content using symmetric encryption and performs key distribution based on homomorphic encryption, and does not need to re-encrypt the content outsourced by content provider. Moreover, our scheme achieves privacy preserving.

Table 3 Comparison with other schemes

DRM scheme	License server	Content encryption	Key distribution	Content re-encryption	Privacy preserving
Scheme in Ref. [5]	Trust	Symmetric encryption	Public key	No	Yes
Scheme in Ref. [11]	Semi-trust	Asymmetric encryption	PRE	Yes	Yes
Our scheme	Semi-trust	Symmetric encryption	Homomorphic encryption	No	Yes

7 Conclusions

In this paper, we first present an efficient digital rights management framework in cloud computing, which allows content provider to outsource encrypted contents to centralized content server, and allows user to consume contents with the license issued by license server. We then provide a secure content key distribution scheme based on AHPE and PRE, which prevents the malicious employees of license server from issuing license to unauthorized user without letting other parties know. In addition, the proposed scheme allows user to stay anonymous towards the key server and service provider, which protects users' privacy.

The analysis and comparison results indicate that the proposed scheme has a lower computational complexity and has high efficiency and security.

Acknowledgements

This work was supported by the National Natural Science Foundation of China (60803157, 90812001, 61272519).

References

1. Feng D G, Zhang M, Zhang Y, et al. Study on cloud computing security. Journal of Software, 2011, 22(1): 71–83 (in Chinese)
2. Taban G, Cardenas A A, Gligor V D. Towards a secure and interoperable DRM architecture. Proceedings of the 6th ACM Workshop on Digital Rights Management (DRM'06), Oct 30–Nov 3, 2006, Alexandria, VA, USA. New York, NY, USA: ACM, 2006: 69–78
3. Qiu Q, Tang Z, Yu Y Y. A decentralized authorization scheme for DRM in P2P file-sharing systems. Proceedings of the 2011 IEEE Consumer Communications and Networking Conference (CCNC'11), Jan 8–11, 2011, Las Vegas, NV, USA. Piscataway, NJ, USA: IEEE, 2011: 136–140
4. Mishra D, Mukhopadhyay S. Privacy preserving hierarchical content distribution in multiparty multilevel DRM. Proceedings of the 2012 World Congress on Information and Communication Technologies (WICT'12), Oct 30–Nov 2, 2012, Trivandrum, IA, USA. Piscataway, NJ, USA: IEEE, 2012: 525–530
5. Wang C K, Zou P, Liu Z, et al. CS-DRM: a cloud-based SIM DRM scheme for mobile internet. EURASIP Journal on Wireless Communications and Networking, 2011(1): 1–30
6. Lee S, Park H, Kim J. A secure and mutual-profitable DRM interoperability scheme. Proceedings of the IEEE Symposium on Computers and Communications (ISCC'10), Jun 22–25, 2010, Riccione, Italy. Piscataway, NJ, USA: IEEE, 2010: 75–80
7. He Y J, Hui L C K, Yiu S M. Avoid illegal encrypted DRM content sharing with non-transferable re-encryption. Proceedings of the IEEE 13th International Conference on Communication Technology (ICCT'11), Sep 25–28, 2011, Jinan, China. Piscataway, NJ, USA: IEEE, 2011: 703–708
8. Corena J C, Ohtsuki T. Secure and fast aggregation of financial data in

cloud-based expense tracking applications. Journal of Network and Systems Management, 2012, 20(4): 534−560

9. Samanthula B K, Howser G, Elmehdwi Y, et al. An efficient and secure data sharing framework using homomorphic encryption in the cloud. Proceedings of the 1st International Workshop on Cloud Intelligence (Cloud-I'12), Aug 31, 2012, Istanbul, Turkey. New York, NY, USA: ACM, 2012: 8p
10. Petrlic R, Sorge C. Privacy-preserving DRM for cloud computing. Proceedings of the 26th IEEE International Conference on Advanced Information Networking and Applications Workshops (WAINA'12), Mar 26–29, 2012, Fukuoka, Japan. Piscataway, NJ, USA: IEEE, 2012: 1286−1291
11. Petrlic R. Proxy re-encryption in a privacy-preserving cloud computing DRM scheme. Proceedings of the 4th International Symposium on Cyberspace Safety and Security (CSS'12), Dec 12–13, 2012, Melbourne, Australia. LNCS 7672. Berlin, Germany: Springer-Verlag, 2012: 194−211
12. Ateniese G, Fu K, Green M, et al. Improved proxy re-encryption schemes with applications to secure distributed storage. ACM Transactions on Information and System Security, 2006, 9(1): 1−30
13. Rivest R L, Adleman L, Dertouzos M L. On data banks and privacy homomorphisms. In: Foundations of Secure Computation. New York, NY, USA: Academic Press, 1978: 169−177
14. Gentry C. Fully homomorphic encryption using ideal lattices. Proceedings of the 41st Annual ACM Symposium on Theory of Computing (STOC'09), May 31−Jun 2, 2009, Bethesda, MD, USA. New York, NY, USA: ACM, 2009:169−178
15. Matt B, Gerrit B, Martin S. Divertible protocols and atomic proxy cryptography. Advances in Cryptology: Proceedings of the International Conference on the Theory and Application of Cryptographic Techniques (EUROCRYPT'98), May 31−Jun 4, 1998, Espoo, Finland. LNCS 1403. Berlin, Germany: Springer-Verlag, 1998: 127−144
16. Ma Z F, Fan K F, Chen M, et al. Trusted digital rights management protocol supporting for time and space constraint. Journal on Communications, 2008, 29(10): 153−164 (in Chinese)
17. Zhang Z Y, Pei Q Q, Ma J F, et al. Establishing multi-party trust architecture for DRM by using game-theoretic analysis of security policies. Chinese Journal of Electronics, 2009, 18(3): 519−524
18. Jiang W, Murugesan M, Clifton C, et al. Similar document detection with limited information disclosure. Proceedings of the IEEE 24th International Conference on Data Engineering (ICDE'08), Apr 7−12, 2008, Cancun, Mexico. Piscataway, NJ, USA: IEEE, 2008: 735−743

(Editor: WANG Xu-ying)