A Trusted Key Management Scheme for Digital Rights Management

Yeonjeong Jeong, Kisong Yoon, and Jaecheol Ryou

ABSTRACT—In this paper, we propose a key management scheme which can provide delivery of the key used to encrypt a digital content from the package server to digital rights management (DRM) clients in a secure manner. The proposed scheme can protect digital content from attacks since an encrypted digital content is sent by a package server and only DRM clients can decrypt the encrypted digital content. It protects the key not only from purchasers but also among the other principals who manage the distribution and license servers.

Keywords—Digital rights management (DRM), key management, content protection.

I. Introduction

Current digital rights management (DRM) provides content protection from purchasers and key management transparently to users. It adopts a license-based mechanism which separates the keys from encrypted contents. The encrypted content is delivered to a player from a distribution server while the license including the keys is transported to the DRM client from a license server. Although the contents are downloaded, users cannot use the contents if they don't have a license [1]-[3].

In Microsoft's DRM, a package server and a license server share a secret seed and can generate their key by using that secret [4]. In a DRM developed by InterTrust, at each moment a package server encrypts a digital content, the package server transmits his encryption key to a license server [5], [6]. Because the license servers of Microsoft's DRM and InterTrust's DRM have or can generate the key used in encryption, it may be possible to extract a raw digital content by using the key.

The current DRM systems have focused on content protection from the purchaser. However, they do not provide a way to protect a digital content from the license server. If the owners of the package server and the license server are not the same, the principal of the package server cannot choose but trust the principal of the license server.

Therefore, we need to provide a way to protect digital content among the other principals who manage the distribution servers and license servers. To protect digital content from attacks, we can use encryption technology. After content is encrypted, only DRM clients can decrypt the content while other clients cannot. This indicates how a content encryption key is protected from its generation to consumption. The key management of a DRM system deals with content encryption, distribution of the content encryption key, delivery of the content encryption key, and authentication between DRM components. In this paper, we propose a key management scheme which can provide secure delivery of the key used to encrypt a digital content from the package server to DRM clients .

II. DRM System

The typical and basic components of a DRM system are a package server, distribution server, license server, and DRM client [1], [2]. The package server encrypts content and generates encryption information such as the seeds of the encryption keys and the encryption length. It provides encrypted content to the distribution server and encryption information to the license server. The distribution server sets up a web site for content and provides encrypted content to the purchaser. The license server issues a license to the DRM client. The DRM client analyzes the

Manuscript received Oct. 04, 2004; revised Nov. 30, 2004.

The third author of this research was supported by the University IT Research Center Project of Korea MIC (the Ministry of Information and Communication).

Yeonjeong Jeong (phone: +82 42 860 6303, email: yjjeong@etri.re.kr) and Kisong Yoon (email: ksyoon@etri.re.kr) are with Digital Content Research Division, ETRI, Daejeon, Korea.

Jaecheol Ryou (email: jcryou@home.cnu.ac.kr) is with the Department of Computer Science, Chungnam National University, Daejeon, Korea.

license and decrypts the content. Figure 1 shows the components of a DRM system and the flow of DRM service among them. The proposed key management scheme will deal with key management among the four components of a DRM system.

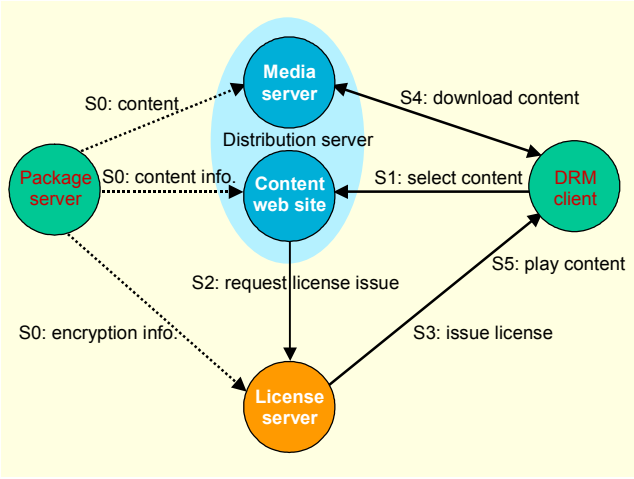

Fig. 1. Service flow in a DRM system.

III. Trusted Key Management Scheme

Based on the results of previous studies [1]-[3], [7]-[11] on a content distribution model and DRM system, we propose a key management scheme in which only package servers and DRM clients can decrypt a digital content while others cannot, one that authenticates DRM clients. Also, we provide a security analysis of the key management in which one can assume certain actions by an adversary. The proposed key management scheme will deal with key management among the four components of a DRM system.

1. Encryption Key

Encryption technology is used to protect data from a user who doesn't have the rights to use them. A symmetric key is usually used to encrypt content because it has a large data volume and needs high performance for encryption. The digital content is encrypted with a symmetric key. For each digital content, a different symmetric key is used to minimize the risk of a key leak.

To protect against the illegal use of symmetric keys, the seeds of symmetric keys are encrypted with a public key and delivered to a principal who owns a secret key. Therefore, the principal who owns a secret key and the seeds encrypted with a public key can generate a symmetric key and decrypt the encrypted digital content.

- A symmetric key is used to encrypt digital content and has seed 1 and seed 2 to generate it.
- An asymmetric key is used to encrypt the seeds of symmetric keys.

2. Key Distribution Scheme

The principals of the package server, distribution server, and license server generate their own secret/public keys and send the public key to the certificate authority (CA) to get a certificate, which is a usual process in a public key infrastructure environment. The components of a DRM system can use the public key, secret key, and public key certificate of the principals. Figure 2 shows the key distribution of asymmetric keys which are used to encrypt the seeds of symmetric keys and mutually used to authenticate the components of a DRM system. To have a certificate issued, the principals of the package server, distribution server, and license server must apply for a certificate at a registration authority (RA) and receive a "reference number" and "authentication code" after the RA verifies the identity of the principals. Then, the principals generate their own secret key and public key and request the certificate of the public key from the CA with the reference number and authentication code sent by the RA. The CA issues the certificate by signing the public key of the principals with the CA's secret key.

The package server issues the certificate of a DRM client to authorize the DRM client but not the purchaser, who is the principal of the IMPRIMATUR business model [7]. There are two reasons why a package server does this. First, the owner of the package server has raw content and wants to protect it.

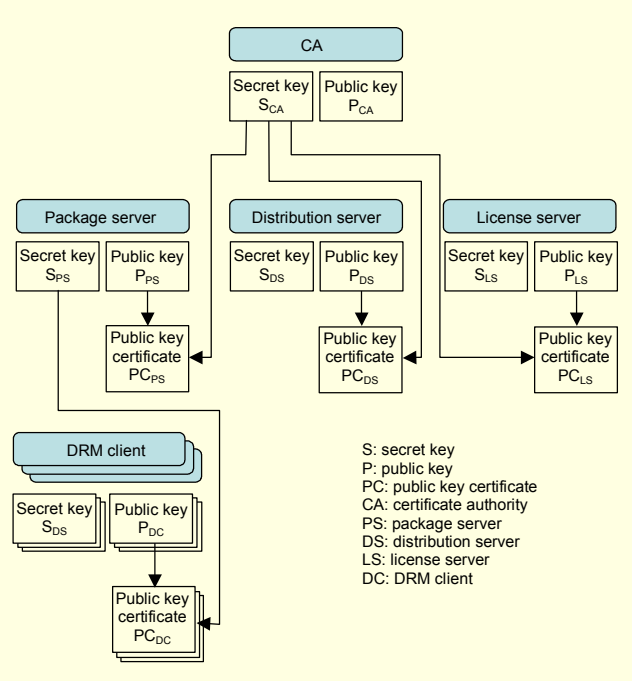

Fig. 2. Asymmetric key distribution.

None of the principals except him should know how to decrypt the content. The components of a DRM system which have a content decryption key are the package server and DRM client. Thus, the owner of the package server needs to authorize the DRM client. Second, it is difficult to authenticate the purchaser and request a public key certificate of the CA from him. He is a customer who simply needs to pay a fee to watch a movie but is not a service provider [12].

3. Key Delivery Scheme

The delivery of key seeds from the package server to the distribution server and license server is described in the following steps. It occurs when the package server packages content. Figure 3 shows the flow of the steps. Because the two key seeds are separated and stored at different servers, neither the distribution server nor the license server can generate a decryption key.

Step 1. The package server verifies the public key certificates of the distribution server and license server.

Step 2. The package server sends a message which contains an encrypted seed to the distribution server and license server after it encrypts one (seed 1) of the seeds with the distribution server's public key and the other (seed 2) with the license server's public key.

Step 3. The distribution server and license server verify the public key certificate of the package server with the public key of the CA. The distribution server and license server verify the message which includes an encrypted seed. Then, each server can decrypt the encrypted seed with its own secret key and store the decrypted seed to its secure database.

The delivery of key seeds to the DRM client is described in the following steps. It occurs to provide a content service to a purchaser when he requests the content service from a distribution server. Figure 4 shows the flow of the steps.

The distribution server and license server can verify whether the DRM client is a legal component to which a seed can be given. The DRM client doesn't have its public key certificate from the CA. However, the distribution server and license server can verify it with the package server's public key according to steps 2, 3, 5, and 6. Because the distribution server encrypts seed 1 with the public key of the DRM client and sends it to the license server, the license server cannot generate a decryption key. The license server also encrypts seed 2 with the DRM client's public key and builds a license which has encrypted seed 1 and seed 2. Thus, the decryption key seeds are decrypted by only one DRM client who has a secret key and not the others.

Step 1. The DRM client sends his public key certificate to the distribution server.

Step 2. The distribution server verifies the public key certificate of the package server with the CA's public key.

Step 3. The distribution server verifies the public key certificate of the DRM client with the package server's public key.

Step 4. The distribution server sends a message which contains encrypted seed 1 to the license server after it encrypts seed 1 with the DRM client's public key.

Step 5. The license server verifies the public key certificate of the package server with the CA's public key.

Step 6. The license server verifies the public key certificate of the DRM client with the package server's public key.

Step 7. The license server issues a license to the DRM client after encrypting seed 2 with the public key of the DRM client. The license contains both encrypted seed 1 and seed 2.

Step 8. The DRM client verifies the public key certificate of the license server with the CA's public key. Then, it parses and analyzes the license which contains two encrypted seeds, rights, content URL, and so forth, and decrypts both encrypted seeds with its secret key. Finally, it obtains a symmetric key which will be used to decrypt the content.

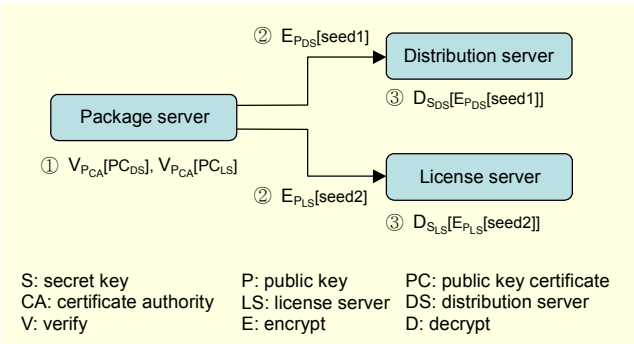

Fig. 3. Key delivery when packaging.

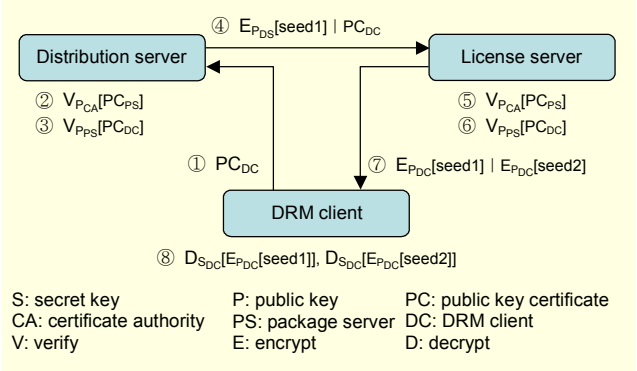

Fig. 4. Key delivery when content service is provided.

4. Analysis

The forgery of a component of a DRM system can be detected by its certificate. One component of a DRM system can authenticate another component by its certificate to determine whether it is a legal component to which a seed can be given. An attack on the two key seeds during delivery from the packager server to the distribution server and the license server is prevented because a seed is encrypted with a public key and delivered to a server who owns a secret key. The two key seeds are separated and stored at different servers; neither the distribution server nor the license server can generate a decryption key by itself. And a decryption key can be protected from an attack on the distribution server or the license server since the two key seeds are separately stored at different servers. Because the distribution server encrypts seed 1 with the DRM client's public key and sends it to the license server, the license server cannot generate a decryption key. It also encrypts seed 2 with the DRM client's public key. Thus, the decryption key seeds are decrypted by only one DRM client who has a secret key and not the others.

IV. Conclusion

Current DRM has focused on content protection from the purchaser. While the key used in encryption is delivered to DRM clients from a package server, it is not protected from the principals who manage the distribution server and license server.

We propose a key management scheme which can provide delivery of the key used to encrypt a digital content from the package server to DRM clients in a secure manner. The key is protected from its generation to consumption. The proposed scheme protects the key from not only the purchaser but also other principals. It can protect the digital content from attacks during the content distribution since the encrypted digital content is sent by the package server and only the DRM client can decrypt the digital content.

References

[1] G Hanaoka, K. Ogawa, I. Murota, G Ohtake, K. Majima, S. Gohshi, K. Oyamada, S. Namba, and H. Imai, "Managing Encryption and Key Publication Independently in Digital Rights Management Systems," *IEICE Trans. on Fundamentals of Electronics, Communications, and Computer Sciences*, vol.E87-A, no.1, Jan. 2004.

[2] Q. Liu, R. Safavi-Naini, and N. P. Sheppard, "Digital Rights Management for Content Distribution," *Proc. Australasian Information Security Workshop Conference on ACSW Frontiers 2003*, vol. 21, Jan. 2003.

[3] J. Lee, S. Hwang, S. Jeong, K. Yoon, C. Park, and J. Ryou, "A DRM Framework for Distribution Digital Contents through the Internet," *ETRI J.*, vol. 25, Dec. 2003, pp.423-436.

[4] Microsoft, http://www.microsoft.com/wondows/windowsmedia/drm/default.aspx.

[5] O. Sibert, "DigiBox: A Self-Protecting Container for Information Commerce," *1st USENIX Workshop on Electronic Commerce*, 1995.

[6] InterTrust, http://www.interturst.com/main/research/index.html.

[7] ISO/IEC, JTC 1/SC 29/WG 11, Information Technology-Multimedia Framework (MPEG-21) - Part 1: Vision, Technologies, and Strategy, N3939, Jan. 2001.

[8] L. J. Camp, "First Principles of Copyright for DRM Design," *IEEE Internet Computing*, vol.7, May-June 2003, pp. 59-65.

[9] D. K. Mulligan, J. Han, and A. J. Burstein, "How DRM-Based Content Delivery Systems Disrupt Expectations of Personal Use," *Proc. 2003 ACM Works. Digital Rights Management*, Oct. 2003, pp.77-88.

[10] M. Valimaki and O. Pitkanen, "Digital Rights Management on Open and Semi-Open Networks," *Proc, WIAPP 2001.* July 2001, pp.154-155.

[11] F. Hartung and F. Ramme, "Digital Rights Management and Watermarking of Multimedia Content for M-Commerce Applications," *IEEE Comm.*, vol. 38, Nov. 2000, pp.78-84.

[12] J. E. Cohen, "DRM and Privacy," *Communications of the ACM*, vol. 46, issue 4, Apr. 2003.