PCMHoDC
A Scheme to Protect Copyright & Modification History of Digital Contents

HeeJae Park and Jong Kim
Department of Computer Science and Engineering
Pohang University of Science and Technology (POSTECH)
San 31, Hyoja-dong, Nam-gu, Pohang, Kyungbuk, South Korea
E-mail: {myphj,jkim}@postech.ac.kr

Abstract: Nowadays, protecting digital contents becomes important because it is easy to copy them and hard to distinguish the copy from the original one. As the Internet becomes wider and faster, digital contents are distributed illegally wider and faster than ever. Much research is conducted on preventing illegal distribution and on developing new protection technologies, such as digital watermarking, digital right management, etc. But these technologies are mainly used for commercial and business purposes. Moreover, these technologies are based on the assumption that digital contents will not be modified after being distributed by the contents owner.

In this paper, we propose a new scheme to protect against the illegal distribution of modifiable digital contents. The proposed scheme also manages modification history and the copyright information of modified digital contents. The proposed scheme assumes that the system is composed of server, client, and application that manages modification history. The application exists in the client side and has a secret key. Any legal user receives encrypted digital contents from the server via this application, but cannot place decrypted contents into storage. If a user has distributed his digital contents and his private key to others, our scheme can determine who has distributed them. We compare our scheme with previous technologies such as simple encryption method, digital watermarking, digital right management, and secure file system, and show that the proposed scheme has better characteristics.

Key words: Copyright protection, management of modification history, modifiable digital contents.

The original version of this chapter was revised: The copyright line was incorrect. This has been corrected. The Erratum to this chapter is available at DOI: 10.1007/978-0-387-35691-4_52

D. Gritzalis et al. (eds.), *Security and Privacy in the Age of Uncertainty*

1. INTRODUCTION

Unlike analogue contents, digital contents, which are represented only by 0 and 1, enables us to achieve a high quality reproduction of original contents. However, digital contents are easy to copy, difficult to distinguish a copy from an original, and hard to protect the copyrights of their creators. Moreover, as the Internet becomes faster and more widespread, illegal copies and distribution occurs more than ever. Although the method of distributing encrypted digital contents to prevent unauthorized copy from illegal users is widely used, it cannot protect the case that a legal user decrypts the encrypted contents and redistributes them as decrypted forms. Therefore, lawsuits about infringements of the copyright of digital contents such as image, audio, video contents, and software programs occur frequently and these litigations of illegal copies show that digital contents have demerits as well as merits.

Methods of protecting the copyright of digital contents have been studied recently. Digital watermarking [1, 15] and digital right management (DRM) [4, 16] belong to them. Digital watermarking technology, firstly applied to multimedia contents such as image, audio, and video files, enables the creators of digital contents to place their copyright information in the contents itself. Digital watermarking studies are expanded to text documents and software programs, but since everyone can view watermarked contents without restriction, this technique does not proactively protect against illegal distribution. Another approach is digital right management (DRM), an integrated technology that guarantees the security of the contents in each step of creation, distribution, and storage. This technique supplies not only the copyright information of digital contents like watermarking techniques, but also mechanisms to prevent the viewing of the contents. A well-known example of using DRM technique is "Windows Media Player" developed by Microsoft [4, 10].

However, DRM focuses only on creation and distribution of the contents but not on their modification. It means that it is impossible to manage the copyright of contents that are modified many times by several people.

In this paper, we propose a scheme, called PCMHoDC, that manages the copyright information of modifiable contents. The proposed scheme manages the modification history and the copyright information of modified digital contents. The proposed scheme assumes a system composed of server, client, and application that manages the modification history. The application exists in the client side and has a secret key. Any legal user receives encrypted digital contents from the server via this application, but cannot place a decrypted one into storage. If a user has distributed his digital contents and his private key to others, our scheme can identify him.

The rest of this paper is organized as follows. In Section 2, we explain the design goals and the system architecture of PCMHoDC. In Section 3, we explain the data formats for digital contents, their communication protocol, and the characteristics of PCMHoDC. We present the previous related works and compare the proposed scheme with them in Section 4. Finally, we summarize in Section 5.

2. DESIGN GOALS AND SYSTEM ARCHITECTURE OF PCMHoDC

In this section, we first present the design goals of PCMHoDC. Next, we show the system architecture and assumptions for PCMHoDC.

2.1 Design Goals

We set three goals related to the protection of modifiable contents, as shown below.

(G1) Authenticated users must be able to cooperate for making digital contents and all participating users must have their copyright.

(G2) Users must be able to modify the contents and must be able to claim the copyright of his modification.

(G3) Secure contents should not be shown to the right-less user and if this occurs, it must be able to identify the illegal distributor.

2.2 The System Architecture and Assumptions

The proposed system architecture is a server-client structure as shown in Figure 1. Assumptions for the server, the MMAP, and a user are as follows.

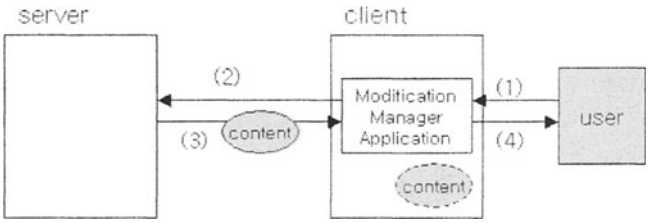

Figure 1: System architecture.

A. Server

- The server does not destruct or remove the contents, and does not distribute them illegally.

- The server encrypts contents with a secret key that the MMAP knows.
- The server has a private key and a public key, and the private key is never disclosed.

B. Modification Manager Application Program (MMAP)

- The server distributes the MMAP to users and users install this MMAP in the client. The MMAP has a secret key unknown to outside of the program by hiding the key in the program itself. This is done by the method of hiding functions [6] or by the data (indistinguishable) obfuscation used for the software protection [7].
- The MMAP stores contents only in the form received from the server. When a user modifies contents, the MMAP sends their modified blocks to the server and receives new version.

C. User

- A user has a private key and a public key and must hide the private key from the public.
- A user can use all resources of client he is connected to.
- A user requests contents to the server, reads and modifies them via the MMAP.

2.3 Notations

- $H(v)$: the hash value of v.
- K_{AppSec} : the secret key of modification manager application program.
- $K_{ser(pr)}$, $K_{ser(pu)}$: a private key and a public key of server, respectively.
- $K_{usr(pr)}$, $K_{usr(pu)}$: a private key and a public key of a user, respectively.
- C_{server} : the form of contents stored in the server
- C_{send} : the form of contents when transmitted.
- C_{client} : the form of contents stored in the client.
- M : the last version of contents.
- M(i) : the i^{th} modified block. Let M_i represent the contents after the i^{th} modification. Then M(i) means the difference between M_{i-1} and M_i.
- $CR_{M(i)}$: the copyright information for the i^{th} modified block. The user making M(i) encrypts the hash value of M(i) and sequence number i of the modified block together with his private key. When the user usrA makes M(i),

$$CR_{M(i)} = K_{usrA(pr)}\{H(M(i)), i\}$$

- D_{id} : the digital contents identifier.
- U_{info} : the information about a specific user.

3. PCMHoDC: PROTECTING COPYRIGHT & MODIFICATION HISTORY OF DIGITAL CONTENTS

In order to maintain the copyright and modification history of digital contents, we propose the data formats of digital contents and communication protocols between the server and the MMAP. Then we explain the characteristics of the proposed scheme.

3.1 The data formats of digital contents

Original contents are stored unencrypted in the server, since it is assumed that the server does not distribute contents illegally. However, when contents are transmitted to a client or stored in the client side, they must be encrypted in order to avoid illegal distribution. So, digital contents have two formats, encrypted and unencrypted.

A. The contents format stored in the server

The server must have contents' identifier, last modified version, modified blocks, and the copyright information of all modified blocks.

- $C_{server} = D_{id} \cdot M \cdot M(n)...M(1) \cdot CR_{M(n)} ... \cdot CR_{M(1)}$

The server stores all contents in the above format. When a user requests them, the server encrypts them and sends them to the user.

B. The contents format transmitted to the client or stored in the client

When a user requests the contents, the server encrypts them with the user's public key, and only that user decrypts them with his private key. After decrypting, a user can know who has modified each part of the contents by seeing the modification information of all modified blocks. There are two methods for receiving the modification information based on when the server sends this to the client.

One method is that the server sends the contents only and does not send the related modification information until a user requests it. Another is that the server sends the contents and modification information simultaneously. While the former has shorter time for viewing the contents, the latter is more effective for viewing the modification information. Table 1 shows C_{send} and C_{client} of each method.

In order to identify the user distributing the contents illegally, user information (U_{info}) is included in the message C_{send}. Also, the hash value of the contents is included to check the alteration of the digital contents and their modification information.

Table 1: The transmission and storing format of digital contents.

		Method1(Contents only)	Method2(Together)
C_{send}	Transmission of contents	$E_{K(AppSec)}\{E_{K(usr(pu))}\{K\},U_{info},D_{id}\}$ $\cdot E_K\{M\cdot H(M)\}$	$E_{K(AppSec)}\{E_{K(usr(pu))}\{K\},$ $U_{info},D_{id}\}\cdot$ $E_K\{M\cdot M(n)...M(1)\cdot$ $CR_{M(n)}...CR_{M(1)}\cdot H(M)\}$
	Transmission of modification information	$E_{K(AppSec)}\{E_{K(usr(pu))}\{K\},U_{info}, D_{id}\}$ $\cdot E_K\{M(n)...M(1)\cdot CR_{M(n)}...CR_{M(1)}$ $H(M)\}$	
C_{client}		Both of the above	The above

3.2 Communication protocols

A user can make and register any contents to the server and request to delete them from the server. And, a user can read and modify the contents by interacting with the server. Therefore, communication protocols between the server and a user must be specified for contents protection when a user wants to register, read, modify, and delete digital contents. These communication protocols must be accomplished under the encrypted mode with their private keys and public keys for security management.

A. When a user requests digital contents for reading.

Figure 2 shows the communication protocol when a user requests digital contents for reading. Each step is described in detail.

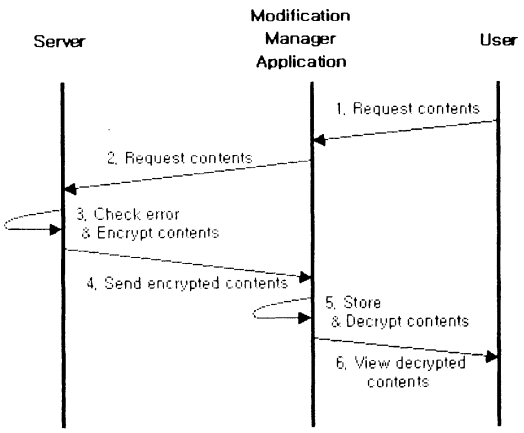

Figure 2: Communication protocol for a user's reading request.

1. A user requests specific contents to an MMAP. He gives his own private key and the identifier of the requesting contents (D_{id}).
2. The MMAP requests the contents to the server.
3. The server sends an error message to the MMAP in the following cases:
   a) The requesting user is not included in the reading list.

b) The requesting MMAP is an illegal copy.
c) The requested contents do not exist in the server.

4. If there exists no error, the server makes the message C_{send} and returns it to the requesting MMAP.
5. The MMAP stores the received, encrypted contents and decrypts them.
6. The MMAP shows the decrypted contents to the user.

B. When a user modifies digital contents.

Figure 3 shows the communication protocol when a user wants to modify digital contents. Each step is described in detail.

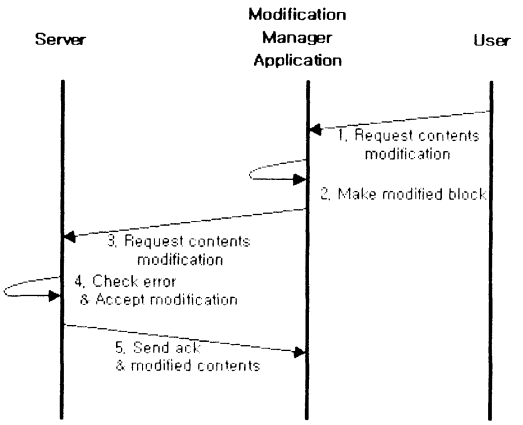

Figure 3: Communication protocol for a user's modification request.

1. A user requests the modification of the contents to the MMAP.
2. The MMAP makes the modified block (M(n+1)) based on the difference between the modified version and the previously received version. It calculates the hash value of this modified block and then makes the copyright information by encrypting the hash value and sequence number (n+1) with the user's private key ($CR_{M(n+1)}=K_{usr(pr)}\{H(M(n+1)), (n+1)\}$).
3. The MMAP encrypts M(n+1) and $CR_{M(n+1)}$ with the server's public key and sends it to the server.
4. After the server checks if the user has the right to modify the contents, it is changed from C_{server}= $M \cdot M(n)...M(1) \cdot CR_{M(n)}...CR_{M(1)}$ to $M \cdot M(n+1) M(n)...M(1) \cdot CR_{M(n+1)}\, CR_{M(n)}...CR_{M(1)}$.
5. The server notifies the MMAP whether the contents are modified successfully and sends their new version to the MMAP.

C. When a user creates new contents.

A user sends new contents to the server by using the same modification protocol shown in Figure 3. At this time, he attaches the list of users with

their rights to read and modify. The server maintains the list and uses it to check the right of another user.

D. When a user requests to delete existing contents.

When the server receives the request to delete contents from the user who have created them, it broadcasts the deletion messages to all users who modified the contents previously. If there is no objection message against it, the server deletes them.

3.3 Characteristics

A. Only legal users can read contents.

The contents are always transmitted and stored in the encrypted form, C_{send} and C_{client}. Since the server sends the contents only to legal users by encrypting with the public key of that user, no one except for the requesting user can see the contents, copyright, and their modification information.

B. The copyright and modification information of digital contents is always managed.

The server stores the copyright and modification information together with the contents as unencrypted because of the assumption that it cannot be cracked. However, it would pose a problem if someone modifies the modification information of contents and distributes it. Therefore, when contents are distributed from the server, it is important to detect the altering of modification information, as well as to keep the contents as encrypted. We consider two cases, shown in Table 1.

In the first method, the server encrypts the modification information together with the contents and distributes in the following data format.

- $E_{K(AppSec)}\{E_{K(usr(pu))}\{K\},U_{info},D_{id}\}\cdot E_K\{M\cdot M(n)\ldots M(1)\cdot CR_{M(n)}\ldots CR_{M(1)}\cdot H(M)\}$

In the above data format, since a user cannot know the random key K as long as the MMAP does not leak its secret key and the key K, he cannot modify the part $E_K\{M\cdot M(n)\ldots M(1)\cdot CR_{M(n)}\ldots CR_{M(1)}H(M)\}$. If a user changes the modification information maliciously, the MMAP can detect it by checking the hash value.

Another method is to encrypt the contents and the modification information separately. In this method, the server distributes the contents and the modification information in the following formats, respectively.

- $E_{K(AppSec)}\{E_{K(usr(pu))}\{K\},U_{info},D_{id}\}\cdot E_K\{M\cdot H(M)\}$

- $E_{K(AppSec)}\{E_{K(usr(pu))}\{K\},U_{info},D_{id}\}\cdot E_K\{M(n)...\ M(1)\cdot CR_{M(n)}...\ CR_{M(n)}\cdot H(M)\}$

The MMAP checks the hash value H(M) whenever it receives the modification information. If a malicious user changes it, the hash value is also changed, and so the MMAP can detect the alteration.

If the MMAP detects the illegal alteration of the modification information, all it has to do is to request the modification information to the server again.

C. Illegal distributor can be found.

Under the assumption that the secret key of MMAP is secure, non-authorized users can view the contents only when an authorized user gives his own private key to them. The proposed scheme has the mechanism of identifying this illegal sharing. The MMAP notifies the user information to the server whenever it requests to read digital contents. The requested digital contents are encrypted with a random key that is again encrypted with the user's public key. Hence, the user's private key is required to see the encrypted contents. If a non-authorized user's reading is detected, the illegal distributor can be determined easily since the user information (U_{info}) is found by the MMAP, by decrypting the contents.

D. It is impossible to modify contents in behalf of another.

We can consider two possibilities.

- When a user wants to modify contents in behalf of another.

  In order for a user to modify contents, he must send the modified block and its modification information, which are encrypted with his private key, to the server. But as a user does not expose his private key, any user cannot encrypt the modification block with another user's private key. Even the replay attack, re-sending the same modified block and the same modification information to the server by eavesdropping, cannot succeed since the server checks the sequence number.

- When the server wants to modify contents in behalf of a user.

  The server may be the dangerous place because contents are stored as unencrypted. It does not destruct any contents, but it may modify some contents as if a user performed the modification. However, the server cannot make M(i) and $CR_{M(i)}$ directly because he does not know a user's private key. The replay attack, using a stored modified block and the matched modification information, can be detected by the sequence number in the copyright information $CR_{M(i)}$ in the client side.

4. RELATED WORKS AND COMPARISON

Similar works related to the proposed scheme are the revision control system, encryption method, digital watermarking, DRM, and secure file system.

First, the revision control system (RCS) has the advantage that it eases the cooperation for modifying contents and controlling the modification history of the contents, and so the study on the revision control via the WWW is proposed recently [8]. However, in RCS, the user who modified the contents cannot have a right to his work and the contents are easy to be disclosed because of no encryption.

The simple encryption method can be used to keep contents secret when transmitting them. Although it enables to distribute the contents and the modification information securely, it does not protect a legal user from distributing the contents in the decrypted form.

Digital watermarking is the technique that inserts special invisible copyright information into the contents itself [1, 2, 3, 15]. In this method, the owner is guaranteed the ownership of the contents, but everyone can view contents because they are distributed without encryption.

Digital right management is the integrated technique that guarantees the right, confidence, security, and the integrity of contents [4, 5, 10, 14, 16]. But this only targets the creation and distribution of contents and does not support modifiable contents.

The techniques of secure file systems support the encryption in file system layer so that only legal users can see the file. Examples are the cryptographic file system [11], capability file names [12], and strong security for distributed file systems [9]. However, re-writing contents to another non-secure file system can make these systems useless.

Table 2: The data formats of other techniques.

	C_{server}	C_{send}	C_{client}
RCS	$M(1)\cdot M(2)\cdot\ldots\cdot M(n)$	$M(1)\cdot M(2)\cdot\ldots\cdot M(n)$	$M(1)\cdot M(2)\cdot\ldots\cdot M(n)$
Encryption Method	M	$E_{K(usr(pu))}\{M\}$	M
Digital Watermarking	M	$M\cdot CR_M$	$M\cdot CR_M$
DRM	M	$E_K\{M\}$	$E_K\{M\}$
Secure File System	$E_K\{M(1)\cdot M(2)\cdot\ldots\cdot M(n)\}$	$M(1)\cdot M(2)\cdot\ldots\cdot M(n)$	$E_K\{M(1)\cdot M(2)\cdot\ldots\cdot M(n)\}$
Bakker's method [13]	M	M	M

Bakker proposed a system preventing illegal distribution in a peer-to-peer environment [13]. This system traces the distribution of all the files in the globe distribution network and when the uploaded software turns out to be

illegal, the system deletes all the files uploaded by that user and bans that user from uploading files. But this system cannot show the owner and the modification information of digital contents.

The data formats of these schemes are summarized in Table 2. RCS manages modified information at any time, and encryption method uses the contents encryption in transmission. Digital watermarking method distributes the contents inseparable from the copyright information, DRM distributes the contents as encrypted and secure file system manages modified information and encrypts it, but reveals the contents as unencrypted when transmitting.

Table 3 shows the comparison of our model with related works. Our scheme has the above four characteristics needed for protecting modifiable digital contents. The encryption method and secure file system have the capability that only legal users can read. The watermarking technique can manage copyright information and digital right management with only these two characteristics. The modification history can be managed in a revision control system. In Bakker's method, only the characteristic of finding the illegal distributor is supported.

Table 3: Comparison with related works.

	Capability that only legal user can read	Management of copyright	Management of modified history	Finding the illegal distributor
Proposed Idea	O	O	O	O
Revision Control System	X	X	O	X
Encryption Method	O	X	X	X
Digital Watermarking	X	O	X	X
DRM	O	O	X	X
Secure File System	O	X	X	X
Bakker's method [13]	X	X	X	O

5. CONCLUSIONS

In the paper, we proposed a scheme, called PCMHoDC, to protect the copyright and modification history of modifiable digital contents. We have shown the architecture, the data format of digital contents, and the communication protocols in order to manage the copyright and modification information. Based on the server-client architecture, we have given the role of protecting the modification to the modification manager application program (MMAP), that guarantees to store contents as encrypted and to show decrypted contents to a user. The server has the role of storing original and modified digital contents. The proposed scheme has more viable characteristics in protecting modifiable digital contents than any other

related works like revision control system, digital watermarking, digital right management, or secure file system, etc.

ACKNOWLEDGEMENTS

This research is supported in part by the Ministry of Information & Communication of Korea, under the "Support Project of University Information Technology Research Center (ITRC)" supervised by KIPA.

REFERENCES

1. M. D. Swanson, M. Kobayashi and A. H. Tewfik, "Multimedia Data-Embedding and Watermarking Technologies," Proceedings of the IEEE, Vol. 86, No. 6, JUNE 1998.
2. J. Brassil, S. Low, N. Maxemchuk, L. O'Gorman, "Electronic Marking and Identification Techniques to Discourage Document Copying," IEEE J. Select. Areas Commun., vol. 13, pp 1495-1504, Oct 1995.
3. Http://www.watermarkingworld.org/webring.html
4. Http://www.microsoft.com/windows/windowsmedia/drm.asp
5. Http://cryptome.org/ms-drm-os.htm
6. T. Sander and C. F. Tschudin, "On software protection via function hiding," Lecture Notes in Computer Science, 1525:111–123, 1998.
7. Boaz Barak, Oded Goldreich, Russell Impagliazzo, Steven Rudich, Amit Sahai, Salil Vadhan, Ke Yang, "On the (Im)possibility of Obfuscating Programs," * Advances in Cryptology - CRYPTO `01, vol. 2139 of Lecture Notes in Computer Science, pp. 1-18, August 19-23, 2001.
8. Jurgen Reuter, Stefan U. Hngen, James J. Hunt, and Walter F. Tichy, "Distributed Revision Control Via the World Wide Web," In Proc. Sixth Intl. Workshop on Software Configuration Management, Berlin, Germany, March, 1996.
9. Ethan Miller, *et.al.*, "Strong Security for Distributed File Systems," the 20th International Performance, Computing, and Communications Conference (IPCCC2001).
10. Microsoft Digital Media Division, "Security Overview of Windows Media Rights Manager," September 2001.
11. Matt Blaze, "A Cryptographic File System for Unix," 1st ACM Conference on Communications and Computing Security, pages 9-16, November 1993.
12. Jude T.Regan and Christian D. Jensen, "Capability file names: Separating authorization from user management in an internet file system," In Proceedings of the 10th USENIX Security Symposium, pages 221-233, The USENIX association, August 2001.
13. A. Bakker, M. van Steen, and A. Tanenbaum, "A Law-Abiding Peer-to-Peer Network for Free-Software Distribution," In Proc. Int'l Symp. Network Computing and Applications, Cambridge, MA, Feb. 2002. IEEE.
14. Http://www.intertrust.com/main/research/whitepapers/IDCUnderstandingDRMSystems.pdf
15. C.I. Podilchuk and E.J. Delp, "Digital watermarking: Algorithms and applications," IEEE Signal Processing Magazine, vol. 18, no. 4, pp. 33–46, July 2001.
16. F. Hartung and F. Ramme, "Digital Rights Management and Watermarking of Multimedia Contents for M-Commerce Applications," IEEE Communications Magazine, vol. 38, no. 11, pp. 78-84, Nov. 2000, Invited paper.