

Digital Rights Management System with Partial Use Property for Home Networks

Narn-Yih Lee and Yun-Hsien Chen

Department of Information Management, Southern Taiwan University,
No. 1, Nan-Tai Street, Yung-Kang City, Tainan County, Taiwan 710, R.O.C.
nylee@mail.stut.edu.tw, N9590011@webmail.stut.edu.tw

Abstract. This paper proposes a simpler and securer architecture of Digital Rights Management for home networks. Besides, our scheme can support partial use of contents and reduce the number of secret keys in the system. Our DRM architecture does not require continuous network connectivity between devices.

Keywords: Digital Rights Management, Digital contents, Home network, Encryption, Cryptography.

1 Introduction

Digital Rights Management (DRM) is a technology for protecting the copyrights of content providers and enabling only designated users to access contents [2][3]. Home networks usually equip with many multiple devices such as PCs, PDAs, and MP3 players; even televisions and DVD players. Supporting all devices is very important for DRM systems. It is expected to support many kinds of different devices and media. Users can conveniently enjoy the digital contents without many limitations [6].

In 2004, Popescu et al. [7] proposed a DRM system for home networks which is authorized domains framework. The centralized design is a novel method which allows device authentication for home network environment. It is based on public key cryptography. Each device has its own public/private key pair. However, there are two weaknesses in Popescu et al.'s system. First, the number of keys is too many. Second, the computational complexity is too heavy.

Therefore, in 2007, Jeong et al. [1] proposed a different DRM system with Popescu et al.'s system. Jeong et al.'s system doesn't need the centralized device and allows the partial use of digital contents for home networks. The partial use means that a content is able to divide into several segments. A user can download some segments of content to his/her device.

For this reason, this paper aims to propose a simpler and securer architecture of digital rights management for home networks. Our scheme is inspired from Popescu et al.'s and Jenog et al.'s systems. In our proposed approach, we will use symmetric key cryptosystem to reduce the number of keys and computational complexity. Furthermore, our system can also support partial use of the contents and

each devices can share digital contents by the Domain Server for home networks. Finally, Our DRM system does not require continuous network connectivity between devices.

The rest of the paper is organized as follows. We briefly sketch overview of the architecture for home networks in Section 2. Next, a DRM system with partial use property for home networks is showed in section 3. Section 4, security analysis and comprehensive comparisons among our system with other DRM systems are presented. Finally, we conclude this paper in section 5.

2 The system model

Figure 1 shows the proposed architecture for home Networks. The architecture consists of the following entities[5][7]:

- n **Content Provider** is the entity that sells and manages the digital contents to consumers. It makes usage rules for contents to facilitate a variety of business models. (E.g. Pay-Per-View, Pay-Per-Used, Subscription, etc.).
- n **Content Manager** is the entity that brings new digital contents into the domain by interacting with content provider.
- n **Domain Server** is the entity that keeps track and manage of all devices in the domain. There is only one domain server per home network.
- n **Device** is the entity that reads and plays digital contents.

As we can see, every device registers to Domain Server before joining the home networks. The Content Manager imports new digital contents into the home networks by interacting with Content Provider. Then device can download the digital contents from the Domain Server.

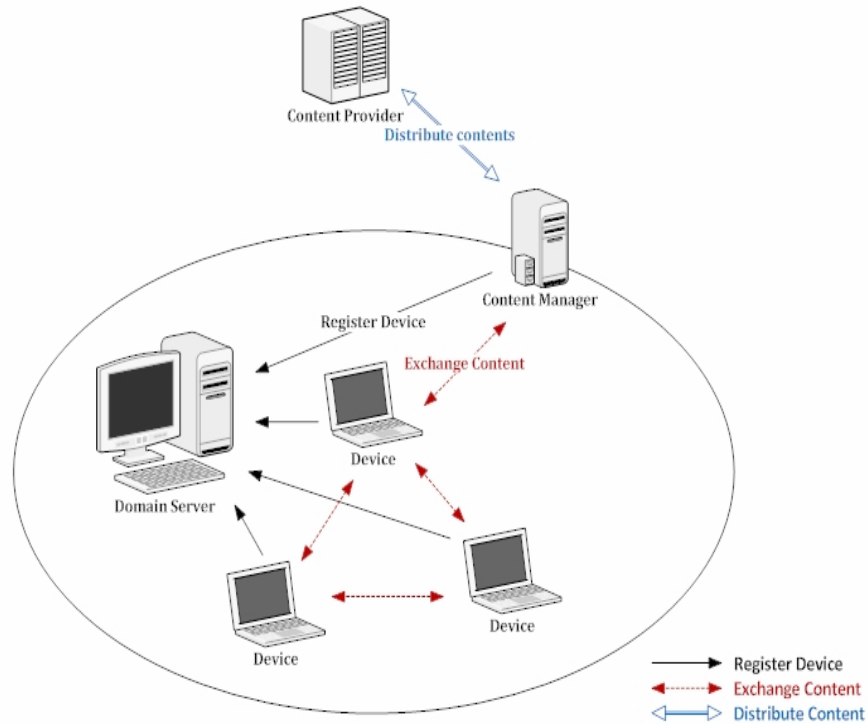


Fig. 1. Proposed DRM Architecture for home networks.

3 The proposed DRM system

In this section, we will present the DRM system for home networks. Table 1 shows the notations used in our system.

Table 1. Notations.

Notations	Descriptions
N_E	A random nonce generated by E
CID	Content identity
DID	Device identity
msg	Device status
$E_X(.)$	Encrypted with the symmetric / asymmetric key X
$H(.)$	Cryptographic one-way hash function
PK_S/SK_S	Asymmetric Key pair owned by Domain Server. SK is private key, PK is the corresponding public key.
R	The amount of Segment

3.1 Six phases in the system

The proposed system consists of six phases: Device registration phase (Fig.2), Upload file list phase (Fig.3), segmented digital contents sharing phase (Fig.4), segmented digital contents selection phase (Fig.5), device login/logout phase (Fig.6), and Upload file variance phase (Fig.7).

A. Device registration phase

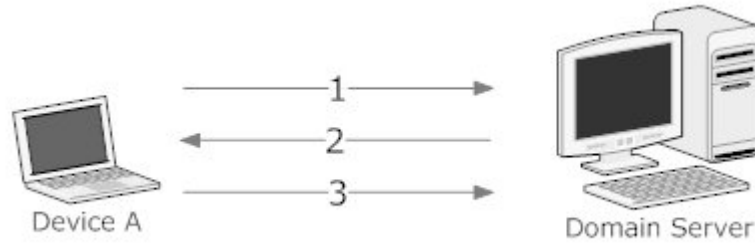


Fig. 2. Device registration phase.

When a new device joins the home network, it must register to the Domain Server.

- Step 1. Device A \rightarrow Domain Server : $E_{PK_S}(DID_A, N_A)$. The Device A uses PKs to encrypt DID_A and N_A , and sends the ciphertext to the Domain Server.
- Step 2. Domain Server \rightarrow Device A : $E_{N_A}(K_A, N_S)$. The Domain Server uses SK_S to decrypt $E_{PK_S}(DID_A, N_A)$ to get DID_A , and calculates $K_A = H(DID_A || SK_S)$. Domain Server uses N_A to encrypt K_A and N_S , then sends it to Device A. K_A does not store in the database of Domain Server. Here, $||$ represents the concatenation of strings.
- Step 3. Device A \rightarrow Domain Server : $E_{N_A}(E_{K_A}(N_S), DID_A), N_S$. The Device A decrypts the ciphertext to get K_A and stores it. Then, Device A uses K_A to encrypt N_S . Finally, Device A uses N_A to encrypt $E_{K_A}(N_S)$ and DID_A , and sends the ciphertext and N_S to Domain Server.

B. Upload file list phase

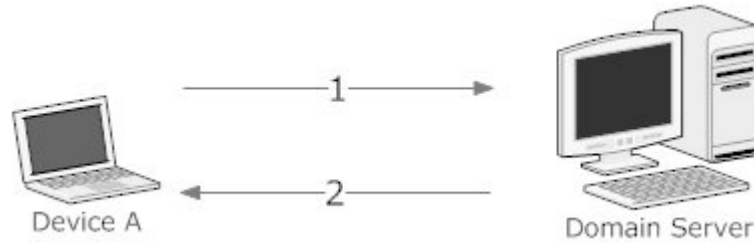


Fig. 3. Upload file list phase.

The Domain Server owns a file list. It records all file names which can be shared with all devices. When a new device registers to Domain Server, the file list will be updated. In this phase, devices will upload file lists to Domain Server.

- Step 1. Device A \rightarrow Domain Server : E_{K_A} (Filename1, Filename2, ..., N_A). The Device A uploads his file list to the Domain Server. He uses K_A to protect it.
- Step 2. When Domain Server receives the package, it will use DID_A and SK_S to calculate $K_A = H(DID_A \parallel SK_S)$ and decrypt it. Finally, the file list will be updated and stored in the database of Domain Server and Domain Server responses a successful message to Device A.

C. Segmented digital contents sharing phase

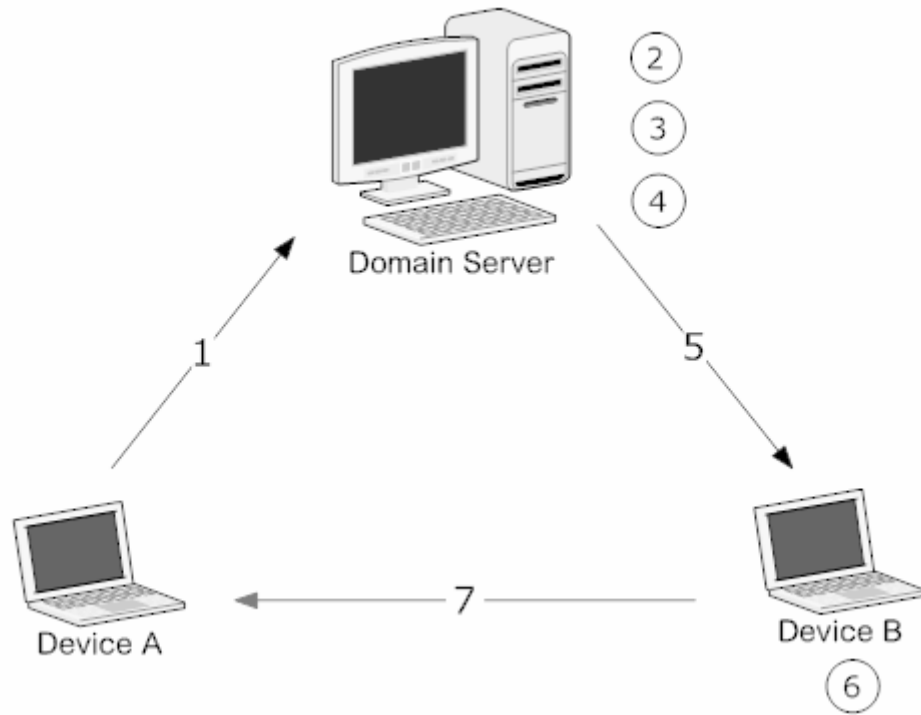


Fig. 4. Segmented digital contents sharing phase.

First, Domain Server selects W_R and calculates $W_R=h(W_{R+1})$, where R is the amount of Segment and $i=0, 1, \dots, R-1, R$. $W_0, W_1, W_2, \dots, W_{R-1}$, and W_R constructs a hash chain. W_0 is called hash chain root. W_0 can be used to verify the correctness of W_1, W_2, \dots , and W_R . In this phase, Device can share digital contents with other devices by Domain Server.

- Step 1. Device A \rightarrow Domain Server : $E_{PK_S}(E_{K_A}(CID, R, N_A), DID_A)$. The Device A used the K_A to encrypt CID, R and N_A . Then, Device A uses Domain Server's public key to encrypt $E_{K_A}(CID, R, N_A)$ and DID_A and sends it to Domain Server.
- Step 2. Domain Server : Compute K_A . The Domain Server calculates $K_A = H(DID_A \parallel SK_S)$.
- Step 3. Domain Server : Search Content. Then Domain Server will search the file list to find the digital contents.
- Step 4. Domain Server : Compute K_B . Assume that Domain Server discovers the content existed in device B, then it calculates $K_B = H(DID_B \parallel SK_S)$.
- Step 5. Domain Server \rightarrow Device B : $E_{K_B}(DID_A, CID, R, N_A)$. The Domain Server uses K_B to encrypt DID_A, CID, R and N_A , and sends the package to Device B.

- Step 6. Device B : $W_R = H(W_R + 1)$, $K_{BA} = H(CID || W_0)$. Device B chooses suitable an R , and calculates K_{BA} . K_{BA} is a private key and does not store in the Device B.
- Step 7. Device B \rightarrow Device A : $E_{N_A}(CID, W_R, R, K_{BA}, DID_B)$. Device B uses N_A to encrypt CID, W_R, R, K_{BA} and DID_B , and then sends the package to Device A.

D. Segmented digital contents selection phase

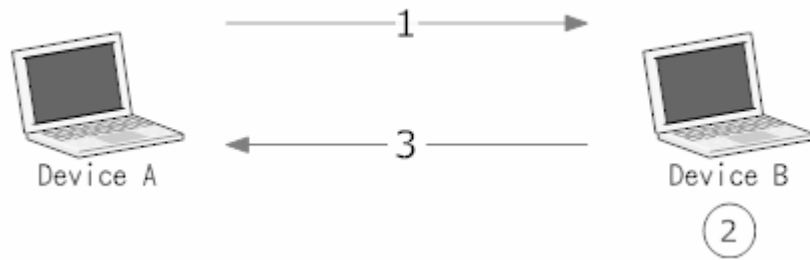


Fig. 5. Segmented digital contents selection phase.

In this phase, users can choose and download segmented digital contents.

- Step 1. Device A \rightarrow Device B : $(E_{BA}(W_2, N_A), E_{K_B}(W_0, DID_A), CID)$. The Device A used E_{BA} to encrypt W_2 and N_A . Then, Device A sends the ciphertext and the package received in the step 7 of the above phase of device B.
- Step 2. Device B : $K_{BA} = H(CID || W_0)$. The Device B uses E_{K_B} to decrypt W_0 and CID , and uses W_0 and CID to calculate $K_{BA} = H(CID || W_0)$. Finally, it uses K_{BA} to get W_2 and N_A .
- Step 3. Device B \rightarrow Device A : $E_{N_A}(Content_2)$. Assume that Device B wants to segment 2 of the content to the Device A. Device B uses the N_A to encrypt it.

E. Device login/logout phase

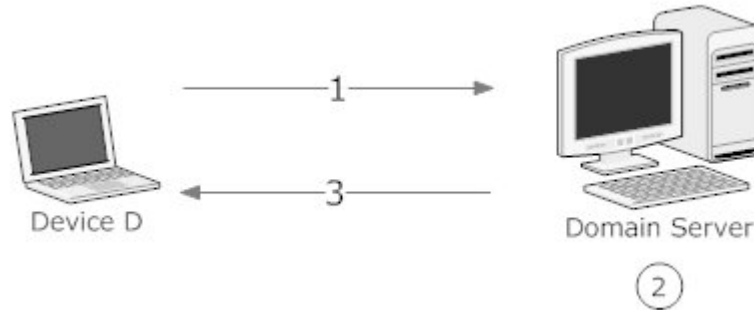


Fig. 6. Device login/logout phase.

Domain Server has a device status table which records current status of devices. In this phase, a device can transfer his current status to Domain Server.

- Step 1. Device D \rightarrow Domain Server : $E_{PK_S}(E_{K_D}(msg, N_D), DID_D)$. The Device D uses the K_D to encrypt msg (current status) and N_D . Then Device D uses Domain Server's public key to encrypt $E_{K_D}(msg, N_D)$ and DID_D . Finally, Device D sends the package to the Domain Server.
- Step 2. Domain Server : Compute K_D . When Domain Server receives the package, he/she will use SK_S to decrypt the package and get DID_D , then calculates the $K_D=H(DID_D \parallel SK_S)$ to decrypt $E_{K_D}(msg, N_D)$ to get msg . Finally, Domain Server can base on msg to update the device state table.
- Step 3. Domain Server \rightarrow Device D : $E_{K_D}(Success, N_D)$. Domain Server updates device state table and responses a successful message to the Device D.

F. Upload file variance phase

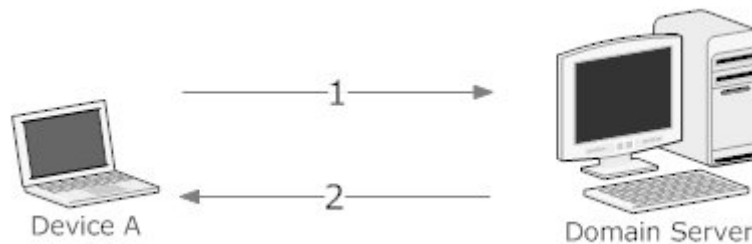


Fig. 7. Upload file variance phase

Device will upload the file variance to Domain Server in a period of time. It is in order to maintain the newest status of files in the database. In this phase, a device can upload file variance to Domain Server.

- Step 1. Device A \rightarrow Domain Server : $E_{PK_S}(E_{K_A}(\text{Filename1, status1, Filename2, status2, } \dots, N_A), \text{DID}_A)$. The Device A uses the K_A to encrypt file variance (filename1, status1) and N_A . Then, Device A uses the domain server's public key to encrypt $E_{K_A}(\text{Filename1, status1, Filename2, status2, } \dots, N_A)$ and DID_A and send it to Domain Server.
- Step 2. Domain Server \rightarrow Device A : $E_{K_A}(\text{Success, } N_A)$. The Domain Server decrypts the package, then the file list will be updated in the database. Finally, Domain Server responses a successful message to Device A.

4 Security analysis and comparisons

4.1 Security analysis

In the device registration phase, if an attacker faked device A to join the home networks, then he/she need to get the DID_A . Therefore, an attack must get the SK_S to decrypt $E_{PK_S}(\text{DID}_A, N_A)$. SK_S is Domain server's private key, so an attack cannot get it, unless the SK_S is exposed. If an attacker want calculates K_A by himself/herself, then he/she needs DID_A and SK_S of device A. But we conclude an attacker cannot get the DID_A and SK_S . Therefore, an attacker cannot fake device A to join home networks.

In the upload file list phase, if an attacker want to eavesdrop the file list, then attacker need K_A to decrypt it. But domain server didn't store K_A in domain server database. Therefore, an attacker cannot get K_A the even if he/she intrude domain server database.

In the segmented digital contents selection phase, the device login/logout phase, and upload file variance phase, if an attacker eavesdrops the information of packet in those phase, therefore he/she need SK_S and K_A to decrypt it. But SK_S is not disclosure and store in the domain server, and K_A is generated by SK_S . For this reason, the information of packet is secure in those phases.

In the segmented digital contents selection phase, if an attacker wants to get W_2 and N_A , he/she must know the key K_{BA} to decrypt the ciphertext. But $K_{BA} = H(\text{CID} || W_0)$, and W_0 is not public. Therefore, an attacker cannot calculate the K_{BA} . Besides, an attacker want to fake device A and send packet to device B, then Device B can use $E_{K_B}(W_0, \text{DID}_A)$ to verify it.

4.2 Comparisons

Here, we compare our DRM system with the Jeong et al [1], and Li et al [4]. The comparisons among these DRM systems are shown in Table 2.

Table 2. The comparisons of some DRM architecture for home Networks .

	Jeong et al.	Li et al.	Ours
Number of Secret Key	2n	2n	2+n
Computation cost	High	High	Low
Encrypted count keys	Chain	Chain	Single
Device authentication	X	X	O
Security	O	O	O
Partial use of contents	O	O	O

5 Conclusions

In this paper, we have proposed a new DRM system to enable contents to be shared partially. Users can share digital contents in many devices freely in the home network. The Domain Server does not store secret key of all devices. Besides, our system can support partial use of digital contents and keeps less secret keys.

References

1. Jeong, E. S., Sur ,C. and Rhee, K. H.: A New DRM System Based on Graded Contents Sharing and Time-Block Distribution for Home Networks. In : Computer and Information Science, pp.830 – 833. July 11-13 (2007)
2. Kalman, G. and Noll, J.: Right management infrastructure for home content. In : Proceedings of the 16th IST Mobile and Wireless Summit, pp.1 – 5. July 1-5 (2007)
3. Kalman, G. and Noll, J.: User Controlled Content Access. In : IEEE Symposium on Wireless Pervasive Computing, Santorini, Greece, pp.185 – 188. May 7-9 (2008)
4. Li, Q., Zhang, J., Gong, X. and Zhang, Z.: A Novel License Distribution Mechanism in DRM System. In: Advanced Information Networking and Applications - Workshops, pp.1329 – 1334. March (2008)
5. Michiels, S., Joosen, K. V. W. and Decker, B. D.: Towards a Software Architecture for DRM. In : Proceedings of the 14th ACM Workshop on Digital Rights Management, pp. 65 – 74. (2005)
6. Nair, S. K., Popescu, B.C., Gamage, C., Crispo, B. and Tanenbaum, A.S.: Enabling DRM-preserving Digital Content Redistribution. In : Proceedings of the Seventh IEEE International Conference on E-Commerce Technology, pp.151 – 158. July 19-22 (2005)
7. Popescu, B.C., Crispo, B., Tanenbaum, A.S and Kamperman , F. L. A. J.: A DRM Security Architecture for Home Networks. In : The 4th ACM Workshop On Digital Rights Management, pp.1-10. (2004)