IDEA GROUP PUBLISHING

701 E. Chocolate Avenue, Suite 200, Hershey PA 17033, USA
Tel: 717/533-8845; Fax 717/533-8661; URL-http://www.idea-group.com

ITP4870

Multiple Rights Management Framework for the Personal Private Networks

Kyung-Ah Chang, Byung-Rae Lee (Samsung Electronics)
Software Center, Samsung Electronics
Apkujung Bldg. 599, Shinsa-dong, Kangnam-gu, Seoul, Korea, 135-893
Tel. (+82)2-3416-0607/ Fax. (+82)2-3416-0302
kachang@samsung.com

ABSTRACT

This paper describes a multiple rights management framework for the multimedia content enforcement. Today, contents sharing and superdistribution in the personal private networks, including home network and office network, has become easy due to advancement in computer technology. However, most DRM server groups only works with their own DRM client, so end-users must have various vendors' clients software to play contents encoded by each vendors' server systems. And the DRM server groups may worry that end-users could give or lend multimedia contents to another domain where the DRM server groups don't allow. To resolve these problems, we propose a new scheme in which multiple rights and contents are managed not by the client in each device but by proxy manager delegated to control the multimedia devices in the personal private networks. This scheme supports rights governance by public key based group communication scheme that ensures that only legitimate operations can apply to the content.

1 INTRODUCTION

Today, contents sharing and superdistribution in the personal private networks, including home network and office network, has become the main area due to advancement in computer technology. This allows end-users to expect the multiple connected storage and display terminals within defined personal private networks, and the peer-to-peer distribution of un-protected and protected contents over the public network.

However, Digital Rights Management (DRM) [8, 9] server groups, consisting of a content provider, a distributor for licenses and contents, and a clearing house, may worry that end-users could give or lend multimedia contents to another domain what the DRM server groups don't allow. DRM systems provide content which is accompanied by rules or controls that define the ways in which the content can be used. The rules are enforced by a governance mechanism that ensures that only legitimate operations can applied to the content. Although DRM server groups adhere to the strict traditional model, end-users want the flexible advancement such as contents sharing with their devices.

Most DRM server groups (e.g. Microsoft, Real networks) only works with their own DRM client. For example, Windows media right manager in server side [18] works with Windows media player and Real system media commerce suite in server side [17] works with Real player.

The challenge of this paper is to efficiently handle event of personal private networks in which end-user must have various vendors' clients on his device and process the multiple rights of dynamic group where wireless devices join and departure the group.

1.1 Our Approach

We propose a new scheme in which multiple rights and contents are managed not by the client, which is a rendering software, in each device but by proxy manager delegated to control the multimedia devices in the personal private networks, such as home network and office network. This scheme supports rights governance by public key based group communication that ensures that only legitimate operations can apply to the content.

Our model assumed that it must be connected through proxy manager between the access network of DRM server groups and the personal private networks. There are all security aware devices, including wireless, that support the dynamic join and departure. In public key based our proposed systems, the proxy manager acts as trusted intermediaries when authenticating devices in the personal private networks. And the DRM smart client must be guaranteed only single embedded client by the tamper resistant software per device.

For our network environment, we follow the security model based on the Universal Plug and Play (UPnP) [10, 14]. Our model is similar to the UPnP architecture for the pervasive connectivity of any device type. It is a distributed, open networking architecture that leverages TCP/IP and web technologies to enable seamless proximity networking in addition to control and data transfer among networked devices.

This paper consists of three parts: multiple rights management framework, mechanism, and conclusion. Section 2 gives the multiple rights management framework and shows the structure of the proxy manager and the DRM smart client for personal private networks. Section 3 explains the multiple rights mechanisms for content rendering with the group key management. Finally, section 4 gives a conclusion and future work.

2 MULTIPLE RIGHTS MANAGEMENT FRAMEWORK

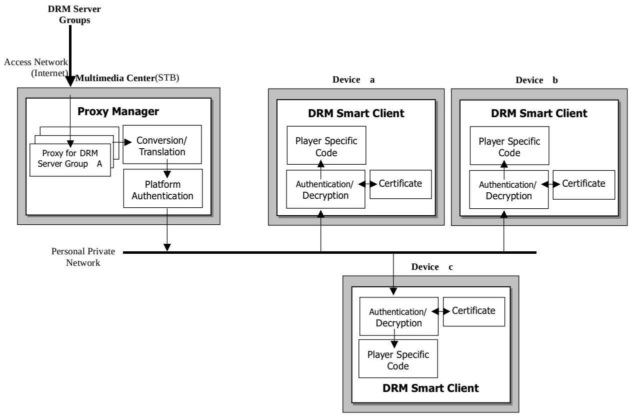
For the purposes of the multimedia content enforcement, a DRM server group requires that only his DRM client be installed on the end-user's rendering device in the personal private networks. Most DRM server groups may worry that end-users could give or lend multimedia contents to another domain where the DRM servers don't allow. Their own DRM solution helps protect the multimedia content by packaging files. A packaged file contains a version of a media file that has been encrypted and locked with a key then can only be rendered by an end-user who has obtained a their own license.

The model described in this paper can be used in the multiple rights management framework based on the proxy manager and the DRM smart client. The proxy manager is responsible for the adaptation of various DRM packaged file by generating of proxy per each DRM server group. The DRM smart client provides services for handling license information and rendering multimedia contents. In this section, we propose the proxy based DRM supporting model concentrated on the personal private networks.

2.1 DRM Service Model

Our DRM model is based on the proxy manager which is similar to the security console in the UPnP. In this model, a networked device

Fig. 1. DRM Service Model



enforces its own right of the multimedia content, but its right control policy is established and maintained using some group key management scheme by a proxy server.

The proxy manager enhances the group key management and the certificate distribution as trusted intermediary, and provides the rights administration and the usage report as security console. The DRM smart client handles protected DRM package and renders the multimedia contents on this device.

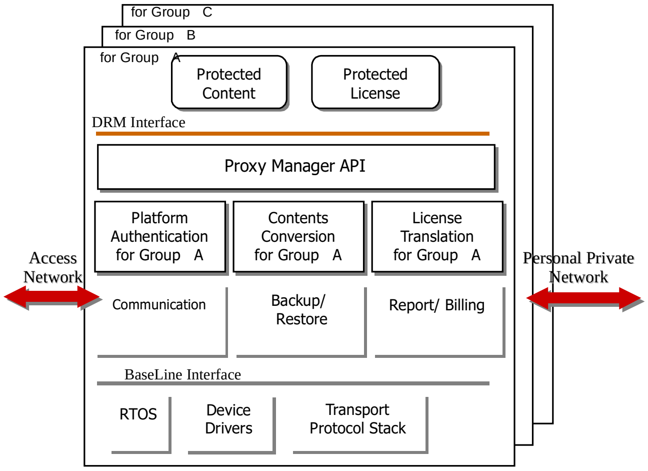
2.2 Proxy Manager

The proxy manager is responsible for adapting various DRM packaged file by generating proxy per each DRM server groups, and for granting process rights to networked devices under its control.

In public key based proposed model, the proxy manager acts as trusted intermediaries when authenticating devices in the personal private networks. General certificate authority issues a long-lived certificate. In the proposed scheme, the proxy manager issues a short-lived credential to devices, which must then be presented to obtain an access right for a particular content in the personal private networks.

The proxy manager is delegated to manage the multiple rights as own clients for various DRM server groups of outer access network, Internet. Most cases it has a local storage, personal digital recorder or hard disk drive, for the backup/ restore of multimedia contents, and supports the computation of strong hash function [11] and random number generation [10, 13] for the key management.

Fig. 2. Proxy Manager Architecture



Main modules of the proxy manager are as follows,

- *Platform Authentication* performs the registration and the authentication of networked devices by the certificate, and supports the right of authorized device for a particular content.
- *Contents Conversion* provides the adaptation of encryption/ decryption method between DRM server groups and the low capacity devices.
- *License Translation* provides the XML-based extraction method between own encryption support for license by DRM server groups and the low capacity support by the networked devices. And later this module supports the report of collecting and billing for the clearing house.

2.3 DRM Smart Client

The DRM smart client provides services for handling license information and rendering multimedia contents. It must be protected by the tamper resistant software against the illegal usage, except authorized pay-per-view services, etc.

To render the DRM-enabled multimedia contents, the end-user needs a media renderer that supports the DRM smart client, and then can render the multimedia contents according to the rules of rights that are translated for the license information. Rights can have different rules, such as start times and dates, duration, and counted operations. Our proposed scheme ensures that the converted multimedia content can only be rendered by the device that has been granted the legal key for that multimedia content.

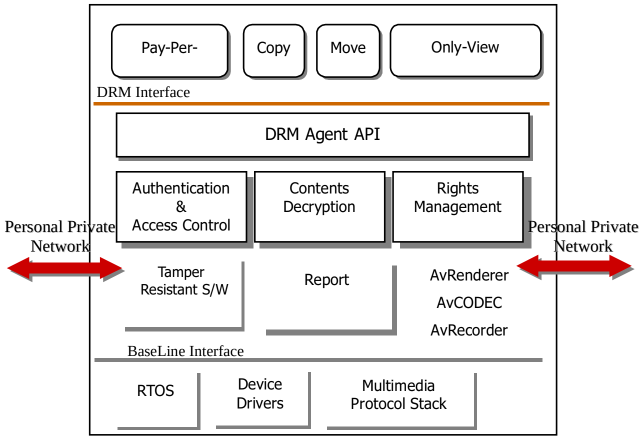
Main modules of the DRM smart client are as follows,

- *Authentication/ Access Control* provides the end-user authentication and the access control: Only access for the authenticated end-user is available to multimedia content services.
- *Contents Decryption* performs the decryption scheme for converted contents and supports the group key management for personal private networks: After the content decryption, it is possible to render the multimedia content on this device.
- *Rights Management* provides an enforcement of the light-weight license information: The right object is made by the license translation.

3 MULTIPLE RIGHTS MANAGEMENT MECHANISMS

We propose a new scheme in which multiple rights and contents are managed not by the clients on each device but by proxy manager

Fig. 3. DRM Smart Client Architecture



which controls the multimedia devices of the personal private networks. This scheme supports rights governance by public key based group communication that ensures that only legitimate operations can apply to the content.

And we consider the following simplified scenario [15, 16]. For the proposed DRM model, a group *g* of *n* devices is receiving encrypted content from proxy server. All devices d_i share a group key k_g which is managed by a proxy server. We assume that the proxy manager can communicate with each client devices using secure one-to-one channels, which are realized issuing the device key k_i of the device certificate. The proxy manager delivers the content encrypted with the group key, to ensure that only authenticated devices can use it.

3.1 DRM Functions

DRM Server Groups Phase. In the access network, DRM server groups regard the proxy manager as the typical client entity and perform general DRM service functions by generating the proxy per each DRM server group on the proxy server. After PKI based long-lived key establishment, DRM server groups perform the authentication of the proxy manager and distribute the protected contents by the various group key managements [1, 2, 6, 7], including the Logical-Tree-Hierarchy (LTH) [4, 5] or One-way Function Trees (OFT) [3], in the access network.

Proxy Manager Phase. Once a proxy manager is activated for DRM server groups, there will be each proxy of DRM server group that grant accesses on multimedia content. It is possible that a proxy manager verifies the devices before service activation of generated proxy for the DRM smart client's request.

If the platform authentication module is succeed, in order to execute the content conversion module and the license translation module, the valid license must be delivered to the proxy server. Then these modules invoke the group key management mechanism which supports the personal private networks.

DRM Smart Client Phase. Before a networked device is required to render the protected DRM content, end-user must prove his identity by the authentication/ access control module, and then it is activated for the DRM smart client to generate a new right session or to verify a certificate of the device. For a right session, the proxy manager assigns the device key then broadcasts this update state in the personal private networks. Once a DRM smart client has established a session, it invokes the signed right from a license translated and decrypts a converted content. This session may be ended intentionally by expiring contents service session keys or by invoking time out at the device's discretion.

Fig. 4. Proposed Multiple Rights Management

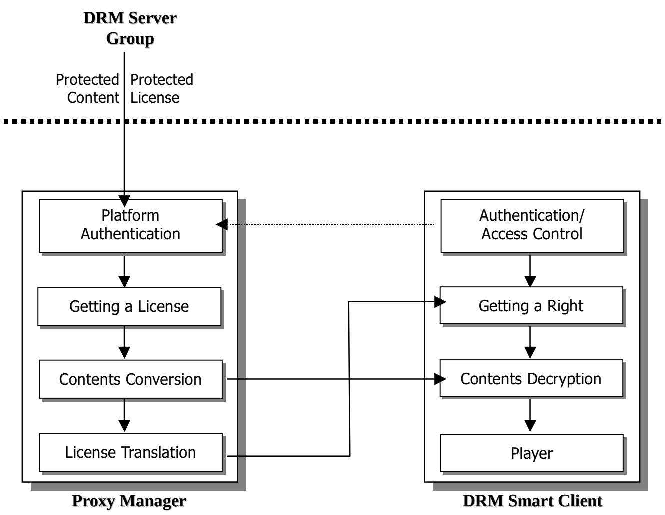
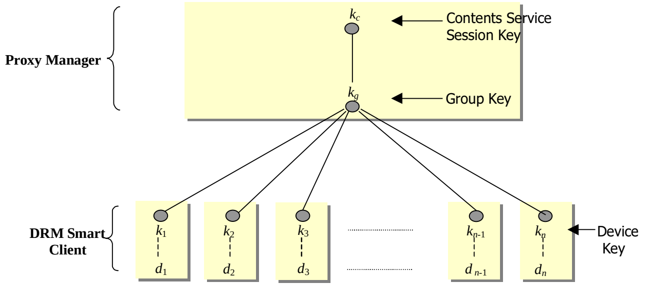


Fig. 5. Group Key Assignment



3.2 Dynamic Group Communications

A simple solution to the personal private networks is based on the single domain multicast model, single proxy manager and multiple devices, where a group of *n* DRM smart clients may share a multimedia content. In this simple tree-based key distribution scheme, device d_i, including wired and wireless, has two key encrypting keys k_i and k_g, and the contents service session key k_c. The contents service session key k_c is used to encrypt the converted multimedia content by proxy server. The k_g is the group key and is used to encrypt messages that update k_c. The remaining keys k_1, k_2, … , k_n are key encryption keys that are used to protect updates of k_g. Both key encryption keys and the contents service session key are assumed to be ^{2} bits in length. We assume the proxy manager makes available a one-way hash function $f(x, y)$ that map sequences of 2^2 bits into sequences of ^{2} bits.

Registration/ Refresh. The device registration step consists of issuing the certificate to establish for the short-lived key, and performing the authentication/ access control. The proxy manager assigns key encryption key k_i of certificate to device d_i.

In order to limit the risk of key compromise, it is necessary to refresh keys prior to key expiration. The refreshing operations will take place during time interval $t – 1$. In the key refreshing stage, it is not necessary to renew $k_g(t) = k_g(t – 1)$. In order to update service session key $k_c(t – 1)$ to a new session key $k_c(t)$, the proxy manager generates $k_c(t)$ and encrypts it using the group key $k_g(t)$. This produces a state update message $m_c(t) = E_{Kg}(t)(k_c(t))$ and is then broadcast.

Device Join. Suppose that during time interval $t – 2$ a new DRM smart client contacts the multimedia content service desiring to become a group member. If there were $n – 1$ DRM smart clients at time $t – 2$ then there will be *n* DRM smart clients at time *t*. During time interval $t – 1$ the rekeying information must be distributed to the $n – 1$ current members. We must renew both the k_c and the k_g in order to prevent the new device from accessing previous service. The first stage requires updating the group key from $k_g(t – 1)$ to $k_g(t)$. Since all of the members at time $t – 1$ share $k_g(t – 1)$, the proxy manager may service the new $k_g(t)$ securely to theses members by forming the message $m_g(t) = E_{Kg}(t)(k_c(t))$. The message $m_g(t)$ is signed and broadcast to all devices. Next, the contents service session key is updated to $k_c(t)$.

Device Departure. Suppose, without loss of generality, that DRM smart client *n* decides to leave at time $t – 2$, then both $k_g(t – 1)$ and $k_c(t – 1)$ must be updated. The group key k_g is updated first, and then used to encrypt the new contents service session key. In order to update $k_g(t – 1)$, the proxy manager first broadcasts a random seed $^2(t)$ then forms $k_g(t)$ and calculates the rekeying message as

$$m_g(t) = k_g(t) + \prod_{i=1}^{n-1} f(d_i, \beta(t))$$

which is signed and transmitted. A legitimate device d_i may restore $m_g(t)$ to get $k_g(t)$ by using $k_c(t – 1)$, extracting message and calculating $m_g(t)(\text{mod } f(d_i, \beta(t)))$. The contents service session key is then updated by $m_c(t) = E_{Kg}(t)(k_c(t))$, signing and broadcasting.

3.3 Remote Domain Extension

When the group domain is much extended, the registered remote network as well as local network approach shall be available. We must consider the balance between computation, communication and storage resources for a cost effective system because the overhead of device departure increase linearly with the devices *n*.

Our remote domain extension is attached to the tree above the root node is the service session key k_c. Each node in the tree is assigned a key encryption key which is indexed by the path leading to itself. When a member moves another subgroup of the remote network, multiple keys become invalidated because that device shares theses keys with other devices. The most efficient approach to updating the key is the bottom-up type from the leaf node to the root. After updating the key encryption keys, the group key $k_g(t)$ can be used to encrypt the new contents service session key $k_c(t)$.

In order to update departure d_{111}, as **Fig. 6.**, the proxy manager is broadcast random seed ${}^{2}(t)$, and then computes the keys in the intersection between the path from their leaf to the root and the path from the removed device's leaf and the root.

$$m_{11}(t) = k_{11}(t) + f(k_{110}(t-1), {}^{2}(t)),$$

$$m_1(t) = k_1(t) + \prod_{j=0}^{1} f(k_{1j}(t), \beta(t)),$$

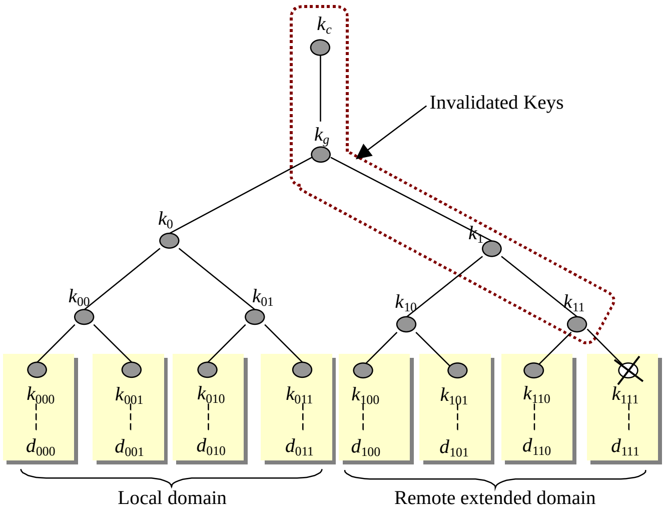
$$m_g(t) = k_g(t) + \prod_{j=0}^{1} f(k_j(t), \beta(t))$$

Finally, the contents service session key is updated by encrypting the new contents service session key $k_c(t)$ using the new group key encryption key $k_g(t)$, and signing the message $m_c(t) = E_{Kg(t)}(k_c(t))$.

4 CONCLUSION AND FUTURE WORK

In this paper, we proposed a new scheme in which multiple rights and contents are managed not by the clients in each device but by proxy manager delegated to manage the multimedia devices in the personal private networks. This scheme supports the rights governance by the group communication that ensures that only legitimate operations can apply to the content and will be the contents sharing and super-distribution in the personal private network, consisting of wired and wireless device.

Fig. 6. Remote Domain Extension



The major direction for continued research should be a formal proof with experimenting the DRM protocol of a wireless network. There might not be continuous communication between the proxy manager and the devices, which can be offline most of the time. This proof should give evidence about the efficiency of the design, and about the implications of the state update [2].

An interesting technical question is to find efficient methods for guaranteeing the legitimate transfer of the protected contents with devices of different domain. Another technical question is the design of an integrated rights management system consisting of several networks and devices, in which only authorized subsets of system can perform transactions, and outer subsets of systems cannot cheat any one of the legal users. This system reduces the amount of trust that is given to multiple group domains.

REFERENCES

[1] A. Fiat and M. Naor, *Broadcast Encryption*, Advances in Cryptology - CRYPTO '93, Springer, LNCS Vol. 773, pp. 480-491, 1994

[2] A. Menezes, P. Oorschot and S. Vanstone, *Handbook of Applied Cryptography*, CRC, 1997

[3] B. Pinkas, *Efficient State Updates for Key Management*, Security and Privacy in Digital Rights Management 2001, Springer, LNCS, Vol. 2320, pp. 40-56, 2001

[4] D. A. McGrew and A. T. Sherman, *Key Establishment in Large Dynamic Groups Using One-Way Function Trees*, Technical Report, No. 0755, TIS Labs (Network Associates), 1998

[5] D. M. Wallner, E. G. Harder and R. C. Agee, *Key Management for Multicast: Issues and Architecture*, Internet Request for Comments 2627, 1999

[6] D. Naor, M. Naor and J. Lotspiech, *Revocation and tracing schemes for stateless receivers*, Advances in Cryptology - Crypto '01, Springer, LNCS, Vol. 2139, pp. 41-62, 2001

[7] J. Anzai, N. Matsuzaki and T. Matsumoto, *A Quick Group Key Distribution Scheme with Entity Revocation*, Advances in Cryptology - Asiacrypt '99, Springer, LNCS, Vol. 1716, pp. 333-347, 1999

[8] J. Duhl, Digital Rights Management (DRM): A Definition, IDC, 2001

[9] J. Duhl, DRM Landscape: Technologies, Vendors, and Markets, IDC, 2001

[10] J. Ritchie, *Media Device Renderer*; ver 1.0, UPnP Forum, 2002

[11] M. Luby, *Pseudorandomness and Cryptographic Applications*, Princeton Computer Science Notes, 1996

[12] M. Luby and J. Staddon, *Combinatorial Bounds for Broadcast Encryption*, Advances in Cryptology - Eurocrypt '98, Springer, LNCS, Vol. 1403, pp. 512-526, 1998

[13] O. Goldreich, S. Goldwasser and S. Micali, *How to construct random functions*, Journal of the ACM, ACM, Vol. 33, No. 4, pp. 792-807, 1986

[14] P. Hunt and U. Warrier, *Using UPnP Technology to Extend the Reach of Handheld Devices*, ver 1.0, Intel Labs, 2002

[15] R. Canetti, J. Garay, G. Itkis, D. Micciancio, M. Naor and B. Pinkas, *Multicast Security: A Taxonomy and Some Efficient Constructions*, Proc. of Infocom, IEEE, Vol. 2, pp. 708-716, 1999

[16] W. Trappe, J. Song, R. Poovendran and K. Liu, *Key Distribution for Secure Multimedia Multicasts via Data Embedding*,. Proc. of Acoustics, Speech, and Signal, IEEE, Vol. 3, pp. 1449-1452, 2001

[17] Real System Media Commerce Suite, Technical White Paper, Real Networks, 2001

[18] Security Overview of Microsoft Windows Media Rights Manager, Andrea Pruneda, Microsoft, 2002

Related Content