

An efficient group rekey scheme for P2P IPTV DRM system

Yuan Zhang*, Yongfeng Huang*, Jian Yuan*

* Department of Electronic Engineering, Tsinghua University, Beijing, China

wanchuangorama@gmail.com, yfhuang@tsinghua.edu.cn, hydrogenleobob@gmail.com

Abstract— This work concentrates on the design of an efficient and scalable group rekey method for P2P IPTV DRM system where the minimal division of right is a program. Our group rekey scheme is efficient in two aspects, less number of encryption and less individual transmission. We divide the members into three categories, namely, the fresh members, the rejoined members, and the remaining members, according to their knowledge to the current content encryption key (CEK) and watching behavior. Different rekey schemes are adopted for each of these categories. For the rejoined members, the number of encryption is reduced from $\Theta(N_{rejoined})$ to $\Theta(k^3 \log N_{rejoined})$, by using broadcast encryption algorithm where k is the collusion size. For the remaining members, a distributed key tree scheme is proposed to allow the remaining members to coordinate a common key. The number of keys encrypted in one rekey process is reduced to $\Theta(k \log N_{remaining})$. To achieve the second goal, we propose a multicast mechanism of P2P network to transmit the new content encryption key for the remaining members once the distributed key tree has been constructed. Individual transmission for the remaining members can be eliminated for the remaining members. This scheme is implemented in our Coolvideo P2P IPTV system. Real implementation suggests that our DRM solution adapts well in P2P IPTV system even when member size grows large.

Keywords— P2P IPTV DRM, Group key management, Watching behaviour, Distributed Key Tree, Broadcast Encryption

I. INTRODUCTION

In the past few years, IPTV application using P2P mechanism has grown significantly. Successful implements such as PPTV, PPSTREAM and UUSEE have been used by numerous people. Each of these applications has attracted thousands of simultaneous users for one hot channel. In a survey conducted in 2009 [1], the peak number of simultaneous users in PPTV exceeds 200,000. Compared to other multicast mechanisms, such as content delivery network, the advantage of using P2P network is that no dedicated infrastructure is needed. Viewers can contribute their uplink bandwidth to a peer-to-peer overlay network and act as relays for media streams.

An easy way to comply with the conference paper formatting requirements is to use this document as a template and simply type your text into it.

With this rapid progress of large-scale IPTV deployments, a demand for copyright protection of P2P IPTV programs has emerged. The general principle of P2P IPTV DRM system is to protect the media content from unauthorized access and unpermitted redistribution. This study has both social and commercial impacts. In social concerns, pay-per-view operation style will leverage the cost of digital right to its end consumer, and let the viewers of P2P IPTV system be responsible for their own consumption. In commercial concerns, the income of advertisement for the P2P IPTV companies hardly covers the cost of TV programs and system maintenance right now [1]. By charging the users for the TV programs, the operation burden of the P2P IPTV companies will be greatly reduced. In a nutshell, although P2P IPTV DRM implementation has not been used in commercial circumstance, the future of the field is promising.

On the other hand, the study of P2P IPTV DRM methods remains roughly untouched. Previous research of this area are [2] and [3]. In [2], C.C. Chu et. al argues that trusted computing techniques, such as client side application and tamper resist hardware can be used in P2P IPTV DRM environment used to enforce digital rights; In [3], X. Liu et.al. provide a framework for P2P IPTV DRM system and discuss the proper media encryption and authentication method. Nevertheless, none of these researches consider the key management scheme in P2P IPTV DRM system, which is the core issue of any secure system.

In P2P IPTV DRM system, the notion of “right” is associated with a particular program within a particular period of time. Users are classified into authorized users and unauthorized users according to their right to the content they required. The requirements of P2P IPTV DRM system can be described as follows:

- Content provider should buy their right before they distribute the media data. To provide copyrighted distribution of media content, the source of the content should be verified.
- The authorized users could be able to watch the media data properly, but they cannot redistribute their rights to others. A single authorized account cannot login in on two terminals at the same time. The unauthorized users and eavesdroppers cannot join the system as a normal member, and even they can obtain the media data by interception, they cannot view it appropriately.

To attain these goals, a set of functions are needed. Figure X illustrates the framework of P2P IPTV DRM system. According to OMADRM standard [4], a DRM system should provide the following functions:

Authorization: users pay to get their rights. The authorization agent gives the paid users some proof to show their right to the resource. This proof can be a public key certificate and/or a record in a trusted secure database. Whatever the form of proof is, it should be able to show the user's authority in the authentication part.

Authentication: the authentication agent verifies the user's authority. In P2P environment, the authentication function may be realized by several distributed authentication agents [5].

Content encryption/decryption: to prevent the media data from eavesdroppers and unauthorized users, the content should be encrypted. The content encryption method should be adapted to facilitate the multimedia being used. A great deal of research has been done in multimedia encryption, among which [6] and [7] are predominate ones.

Group key management: the content encryption key (CEK), which is also referred as the group key in [8], is the key used to encrypt the media data. CEK should be accessible to the authorized users only. The aim of the group key management is to distribute the CEK among all the authorized users, and insure that no unauthorized users can access it. When the authority of user has changed, the CEK should be renewed. This process is called the group rekey.

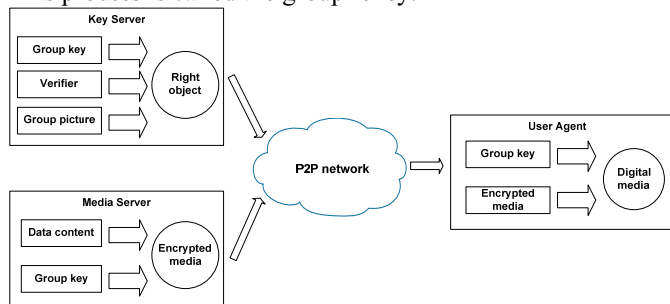


Figure 1. P2P IPTV DRM architecture

However, existing group key management protocols cannot be directly adapted to P2P IPTV DRM system. In P2P IPTV DRM system, the minimal division of copyright is a program. The content encryption key should be renewed between two consecutive programs to a huge amount of users. In the P2P IPTV system, mass peers tend to join and leave within a relatively short period of time, and this dynamic behavior will become even more severe when the customers have to pay for their programs. The sharp dynamic of peers and limit rekey time jointly pose a big challenge for key management in P2P IPTV DRM system. Current group key management schemes generally focus on two directions: the first direction is for the dynamic peer groups [9], it intends to provide group key management protocol for collaborative and interactive small groups, such as teleconferencing, online games, shared instruments, and command-and-control systems; the second direction is for large groups without much dynamics [10] [11], such as cable television or view-on-demand systems. Current

DRM system often suits the second class; nevertheless, P2P IPTV DRM system cannot directly use the existing group key management protocols for large groups. The problem of P2P IPTV DRM system design turns into the problem of key management protocol development. New group rekey methods should be designed for P2P IPTV DRM scenario.

This paper solves P2P IPTV DRM group key management problem by providing a quick rekey scheme for large group. We solve the problem from two approaches: reducing the number of encryption and reducing the individual transmission in the rekey process. The reason why P2P IPTV group rekey is difficult is because peer dynamics. In order to reduce the number of encryption, the members are divided into three categories, namely, the fresh members, the rejoined members, and the remaining members according to their knowledge to the current content encryption key and watching behavior. A specific group rekey scheme which is best for a particular type is adopted for each of them. For the fresh members, the content encryption key is generated immediately after authorization, which provides an initial start point for the fresh members. For the rejoined members, broadcast encryption is used to prevent the revoked users from accessing the new content encryption key and reduce the number of encryption. For the remaining members, a distributed key tree scheme is proposed to allow the remaining members to coordinate a common key among them. By using the distributed key tree scheme, the number of keys encrypted in one rekey process is reduced to k , where k is the size of the remaining members. To reduce the individual transmission, we provide a multicast mechanism to transmit the new CEK for the remaining members once the distributed key tree has been constructed. Individual transmission for the remaining members can be eliminated for the remaining members. This scheme is implemented in our *Coolvideo* P2P IPTV system. Real implementation suggests that our DRM solution adapts well in P2P IPTV system even when member size grows large.

The other parts of this paper are recognized as follows: section 2 briefly reviews previous approaches on group key management. Section 3 describes the motivation and specifications of member classification. Section 4 describes the rekey methods used for each of these three categories and evaluates the rekey method respectively. Section 5 discusses the implementation issues and conclusions are presented in section 6.

II. RELATED WORK

There are several different approaches to efficiently manage a key within a group of members. One is the broadcast encryption approach [14]. A broadcast encryption scheme focuses on reducing the number of encryption and protects the content encryption key from the revoked users. It takes the set of revoked users and the CEK k to output a broadcast message M . An authorized user runs the decryption algorithm that will compute the k associated with M .

Another approach is the group rekey methods. A group rekey method generally focuses on efficiently redistribute the CEK when authority of users changes. There are two types of

group rekey methods: centralized group rekey and decentralized group rekey. In centralized group rekey methods, a key server maintains a CEK with each authorized member in order to enable secure communication between a central resource and all the authorized users. The centralized method works well in one-to-many multicast scenarios, existing DRM applications, such as cable television [12] and view-on-demand system can often use the centralized group rekey methods. [13] is the predominant centralized group rekey method. It uses a directed key tree structure called the Logical Key Hierarchy (LKH) to provide a tradeoff between the number of keys kept by a user and the number of keys encrypted by a server in the rekey process. Each user maintains keys, and the server encrypted keys in the rekeying process. In contrast, decentralized group key distribution methods involve dynamically selecting a group member to generate and distribute keys to other group members. This kind of schemes is designed for many-to-many groups such as teleconference and collaborative online games. The prevalent decentralized group rekey scheme is the one proposed in [9]. It combines a distributed group key management scheme with Diffie-Hellman key exchange algorithm. No root node is needed in their scheme, and rekey is conducted by exchange messages between peer members. The disadvantage is that peer control information must be generated when membership changes to provide the topology discovery and mutual authentication.

A significant different between broadcast encryption method and group rekey method is that, group rekey methods can only be applied to the users who are already in the system (which is noted as the remaining members in the paper) while broadcast encryption methods can reduce the number of encryption even if the user is not in the system. However, the drawback of broadcast encryption method is that it is more complex in terms of computation time and has the danger of collusion attack.

III. GROUP KEY MANAGEMENT ALGORITHMS

A. Member Classification and Motivation

In this section, we describe the group key management algorithms for P2P IPTV DRM system. The motivation and benefits of member classification in P2P IPTV DRM system are discussed. Rekey algorithms of each of the three types of users are presented and the analysis of the algorithms is conducted. After that, we provide a group key multicast scheme in P2P IPTV system.

In P2P IPTV DRM application, the new CEK should be distributed before the next program begins. The short rekey time make the existing group key management schemes infeasible in P2P IPTV application. However, there are two properties can be used to alleviate the difficulty of group rekey.

For the users who are already in the system and have the right of the next program (referred to as the remaining users in this paper), the aim of group key management is to redistribute the CEK for the next program. For this kind of users, group rekey can be done during the current program

rather than in the program interval. Furthermore, the centralized group rekey methods can be used for the members who are already in the system. By using our gradual LKH scheme, the number of key encryption can be reduces from $\Theta(N_{remaining})$ to $\Theta(\log N_{remaining})$. And since the users are

already in the system, multicast techniques can be used rather than individual transmission. This type of member is called the remaining user in the paper. These three factors greatly alleviate the burden of group rekey to the users who are already in the system.

For the users who have some knowledge of the current CEK but are not present in the system, the aim of group key management is to efficiently distribute the current CEK to them. Broadcast encryption methods can be used to reduce the number of keys encrypted in the rekey progress. The number of keys encrypted in the rekey progress can be reduced from $\Theta(N_{rejoined})$ to $\Theta(\log N_{rejoined})$.

Another kind of users is the fresh members, who are not in the system nor have any knowledge of the current CEK. Since the user has no knowledge of the current CEK, no group key management algorithm can be used to facilitate the group rekey process. When this kind of users joins the system, the key server has to individually transmit the encrypted current CEK to them.

B. The Fresh Members

The fresh members are the users who do not have the knowledge needed to compute the CEK for the next program. In this case, the key server has no way but to encrypt the CEK by the individual key of the fresh member and individually transmit the encrypted key to the fresh member. The number of encryption and individual transmission is N_{fresh} , which is the total number of fresh members.

C. The Fresh Members

The rejoined members are the users who have some knowledge of the current CEK but are not present in the system. For the rejoined members, the aim of group key management is to efficient distribute the current CEK to the user. The user's knowledge of the current CEK can be used to facilitate the distribution.

Broadcast encryption methods can be used to reduce the number of keys encrypted in the rekey progress. For the rejoined users, the revoked users can collude to decode the current CEK. The number of keys encrypted in the rekey progress can be reduced from $\Theta(N_{rejoined})$ to $\Theta(\log N_{rejoined})$.

D. The Fresh Members

1) Review: Logical Key Hierarchy: The key server maintains a directed key tree to distribute the group key to the privileged group members. A key tree is a directed tree in which each node represents a key. The root of the key tree is the *group key*, and a leaf node is a user's *individual key*. A user u is given key k if and only if there is a directed path from its individual key to key k in the key tree. An example

key tree is shown in Figure 2. In this group, user u_9 is given three keys: k_9 , k_{789} and k_{1-9} . Key k_9 is the user's individual key, key k_{1-9} is the group key, and k_{789} is an auxiliary key shared by u_7 , u_8 and u_9 .

Suppose u_9 leaves the group. The key server will then need to change the keys that u_9 knows: change k_{1-9} to k_{1-8} , and change k_{789} to k_{78} . To distribute the changed keys to the remaining users, the key server constructs the following *rekey message* by traversing the key tree bottom-up: $(\{k_{78}\}_{k_7}, \{k_{78}\}_{k_8}, \{k_{1-8}\}_{k_{123}}, \{k_{1-8}\}_{k_{456}}, \{k_{1-8}\}_{k_{78}})$. Here $\{k'\}_k$ denotes key k' encrypted by key k . Upon receiving this message, a user extracts the encrypted keys that it needs. For example, u_7 only needs $\{k_{1-8}\}_{k_{78}}$ and $\{k_{78}\}_{k_7}$.

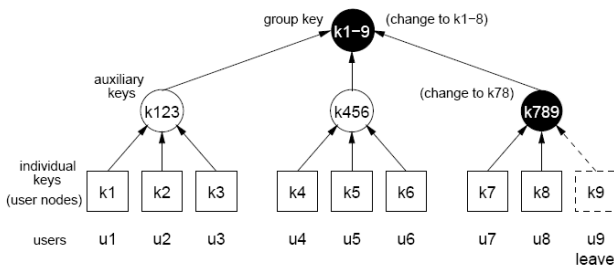


Figure 2. Aggregated key tree structure

2) Problems and Solution: LKH scheme is a general principal for group key management. However, it's not directly applicable for P2P IPTV DRM application. The problem is that there are so many rekeys in the system: every time a node joins or leaves the group, the group key is renewed. Since the nodes can arbitrarily join or leave in the P2P IPTV system, the number of rekeys could be huge. This huge number of rekeys might not be a serious problem in other group communication systems, but it is unbearable in the live streaming system. Unlike the delay tolerant system, in a live streaming multicast application, the delivery of the group rekey must keep in pace with the media data. Once a rekey message is lost or delayed, the loss of the media content cannot be compensated. Fortunately, there are some intrinsic properties in IPTV DRM system that can help to tackle this problem, or at least to reduce its intensity. Here are two special properties:

First, in a live streaming system, the amount of previous media data in a live streaming system is limit. A new join peer's knowledge of backward information is scant even though it can access the previous media data. According to [], the average difference of received media time in an average P2P IPTV system is no longer than 1 second. In this sense, the backward rekey for a new joined user is unnecessary in a P2P IPTV DRM system.

Second, like other DRM systems, the valid time of a peer in the P2P IPTV DRM system is determined at the time the user registers to a trusted registrar. This knowledge is known and could be used afterwards. Forward rekey for a left user is unnecessary if the user is still in its valid time.

3) Program division and minimal length: The concept of the minimal time is important to the aggregated rekey scheme. However, the determination of the minimal length is not simple. On the one hand, a leave rekey should be able to complete during the minimal length. Hence, another rekey can happen in the next minimal length. On the other hand, the duration of the minimal time should not be too long, because all events are assigned on the integral multiple of the minimal length. In this paper, we define the minimal length to 1 minute.

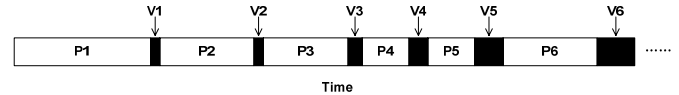


Figure 3. Program division

4) Aggregated Key Tree Structure: We illustrate the architecture and specification of the aggregated key tree structure in this part. In an aggregated key tree, peers with the same relative expiration time are aggregated under a common supernode. The relative expiration time of a node to a program is defined as XXX. The expiration time of the supernode is defined as the relative expiration time of peers under that supernode. Figure X is an example of the aggregated key tree. As can be seen from the figure, an aggregated key tree contains two parts: a supernode key tree and several individual key trees. The supernode key tree has the same structure as a LKH where the supernodes are viewed as leaf nodes. All the supernodes are located in a sorted order. This arrangement can further reduce the number of encrypted keys by diminishing the chaos of random peer positioning. An individual key tree consists of user nodes of the same expiration time, all rooted under a common supernode. The manner of how the users are arranged under the supernode, however, is not constrained. Fig X shows two types of individual key tree under a supernode S. In the first individual key tree, users are directly rooted under the supernode S. In the second individual key tree, the users are organized under auxiliary key nodes. Each of the two types of individual key tree has its pros and cons. In the first individual key tree, users under the same supernode share the same group key and auxiliary keys. The individual key tree does not contain any auxiliary key, so the rekey message contains fewer encrypted keys. However, if revocation happens, the rekey of the individual key tree is inefficient. The key server has to individually encrypt and transmit the supernode key to the remained users under that individual key tree. In the second individual key tree, more keys are kept by the user node, and the rekey message also contains more encrypted keys. However, the rekey for a revocation case is scalable, being exactly the same to an ordinary leave rekey.

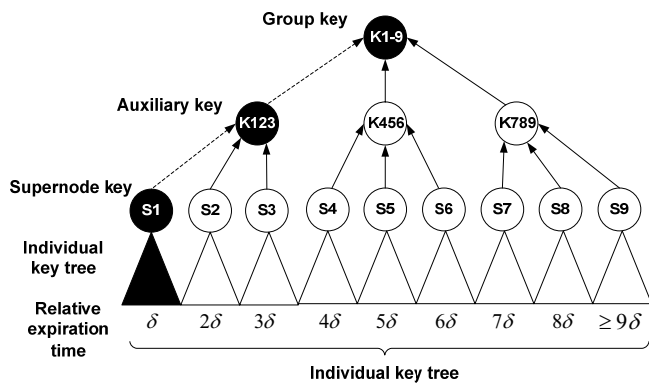


Figure 4. Aggregated key tree structure

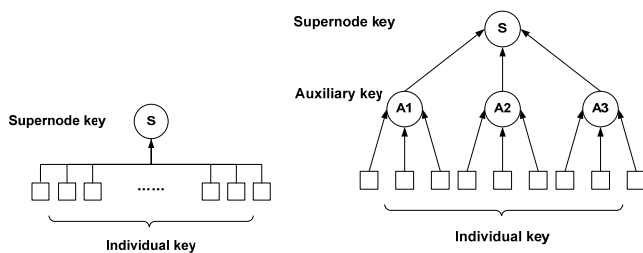


Figure 5. Two types of individual key tree

5) Rekey Algorithm: In the aggregated rekey scheme, a key server maintains a copy of the key tree topology. When rekey is needed, the key server first renews the key tree topology, then it encrypts the keys to be changed and multicasts the rekey message to the group members. A user node is denoted as a NIL node after its leave. For other nodes (the root node or the internal nodes) in the key tree, a node is NIL means all the user nodes under that node are NIL. As in [19], the NIL nodes are used to diminish the change of user IDs in the system. At any time, all the supernodes are used in the AKT. If no users under that supernode, it is denoted as NIL. This skill makes it convenient for the join process and prevents churns caused by change of tree height. Besides leave rekey, another kind of rekey, denoted as the shuffle rekey is also necessary in a system with consecutive TV programs. It constructs the new aggregated key tree for the next program. Users are arranged according to their relative expiration time to the next program. The shuffle rekey happens during the interval of two successive programs and thus does not incur additional burden to the GKM system. Rekey messages for shuffle rekeys contain all the keys needed by the new joined users and the keys changed for the remained ones.

IV. IMPLEMENTATION

This scheme is implemented in our *Coolvideo* P2P IPTV system. Real implementation suggests that our DRM solution

adapts well in P2P IPTV system even when member size grows large.

V. CONCLUSIONS

We propose an efficient group key management scheme for P2P IPTV DRM application by dividing the members into three categories and applying different rekey schemes for each of these categories. The number of encryption and individual transmission are greatly reduced by using our scheme.

REFERENCES

- [1] Report of the China online video industry , 2008-2009, <http://report.iresearch.cn/Reports/Charge/1262.html>
- [2] C.C. Chu, X. Su, B.S. Prabhu, R. Gadh and S. Kurup, "Mobile DRM for multimedia content commerce in P2P networks", *Consumer Communications and Networking Conference*, 3rd IEEE Vol. 2, 8-10 pp. 1119 – 1123, Jan. 2006.
- [3] X. Liu, T. Huang and L. Huo, "A DRM Architecture for Manageable P2P Based IPTV System", ICME 07.
- [4] Open Mobile Alliance Digital Rights Management Standard, version 2, Mar. 2006.
- [5] A. Perrig, R. Canetti, J. Tygar and D. Song, "Efficient and Secure Source authentication for multicast", *Network and Distributed System Security Symposium*, NDSS'01, Feb. 2001
- [6] I. V. Bajic and J. W. Woods, "Domain-based multiple description coding of images and video," *IEEE Trans. Image Process.*, vol. 12, pp. 1211–1225, Oct. 2003.
- [7] Y. Wang and S. Lin, "Error resilient video coding using multiple description motion compensation," *IEEE Trans. Circuits Syst. Video Technol.*, vol. 12, pp. 438–453, Jun. 2002.
- [8] *Group key management protocol (GKMP) specification*, RFC2093, 1997.
- [9] S. Banerjee and B. Bhattacharjee, Scalable secure group communication over IP multicast. JSAC Special Issue on Network Support for Group Communication, 2002,20(8):156–163.
- [10] M. Setiner, G. Taudik and M. Waidnet, "A new approach to group key agreement," *Technical Report*, RZ 2984, IBM Research, 1997.
- [11] W. Diffie and M. E. Hellman, "New directions in cryptography," *IEEE Trans. on Information Theory*, 1976,IT-22(6): pp. 644–654.
- [12] *Specifications of conditional access system for digital television broadcasting*, GY Std, GYZ175-2001, Sep. 2001
- [13] D. Wallner, E. Harder, Agee R. *Key management for multicast: Issues and architectures*, RFC 2627, 1999.
- [14] A. Fiat and M. Naor, "Broadcast Encryption", *Advances in Cryptology: CRYPTO1993* (LNCS773), pp. 480–491, 1993.
- [15] M. S. Iolus, "A framework for scalable secure multicasting," *ACM SIGCOMM Computer Communication Review*, Vol. 27, Issue 4, pp. 277–288, 1997.
- [16] M. Waldvogel, G. Garonni, D. Sun, N. Weiler and B. Plattner, "The VersaKey framework: Versatile group key management," *IEEE Journal on Selected Areas in Communications (Special Issue on Middleware)*, 1999,17(9), pp. 1614–1631.
- [17] D. Balenson, D. McGrew and A. Sherman, *Key management for large dynamic groups: One-Way function trees and amortized initialization*, IETF Internet Draft (work in progress), 2000.
- [18] X. Hei, C. Liang, J. Liang, Y. Liu, and K. W. Ross, "Insights into pplive: A measurement study of a large-scale P2P IPTV system," *Proc. IPTV Workshop Conjunction WWW*, May 2006.
- [19] Y. R. Yang, X. S. Li, X. B. Zhang, S. S. Lam. "Reliable group rekeying: A performance analysis," *ACM SIGCOMM 2001*. San Diego, 2001.
- [20] O. Rodeh, K. Birman and D. Dolev, "Optimized group rekey for group communication systems," *Technical Report*, Hebrew University, 1999.