國立交通大學

資訊管理研究所

碩 士 論 文

一個針對IPTV服務以頻道分群為基礎的金鑰管理協定

A Channel-based Key Management Protocol for IPTV Services

研 究 生： 李孟儒

指導教授： 羅濟群 教授

中華民國 壹 百 年 五 月

一個針對IPTV服務以頻道分群為基礎的金鑰管理協定

A Channel-based Key Management Protocol for IPTV Services

研 究 生：李孟儒　　Student: Meng-Ju Lee

指導教授：羅濟群　　Advisor: Chi-Chun Lo

國 立 交 通 大 學

資 訊 管 理 研 究 所

碩 士 論 文

A Thesis

Submitted to Institute of Information Management

College of Management

National Chiao Tung University

in partial Fulfillment of the Requirements

for the Degree of

Master

in

Information Management

May 2010

Hsinchu, Taiwan, Republic of China

中華民國 壹 百 年 五 月

一個針對IPTV服務以頻道分群為基礎的金鑰管理協定

研究生：李孟儒　　指導教授：羅濟群

國立交通大學資訊管理研究所

摘要

隨著網路快速發展，IPTV (Internet Protocol Television)將成為一個可行的服務。IPTV 需要一個安全、有效率、且具有延展性的金鑰管理協定，來防止未授權的使用者看到 IPTV 傳播之內容。就此議題，本論文針對 IPTV 服務提出一個以頻道為分群基礎的金鑰管理協定。於此協定中，IPTV 服務提供者握有多把群組金鑰、頻道金鑰、輔助金鑰及密鑰；訂閱者握有一把群組金鑰、多把頻道金鑰、輔助金鑰及密鑰。此外，此協定提出金鑰更新演算法，包括：定期更新金鑰、使用者訂閱、不訂閱 IPTV 服務、或當使用者改變訂閱 IPTV 服務內容演算法。本論文亦提出樹平衡演算法以維持金鑰更新效率。我們對所提出的協定與其他相關協定進行分析：從安全分析上，本協定多提供向前安全(forward secrecy)、向後安全(backward secrecy)，並且避免共謀攻擊(collusion attacks)的發生；在效率分析上，本論文以傳送的訊息數量、服務提供者的運算量、以及金鑰的儲存成本來做比較。本論文所提出的協定，雖然在儲存成本上高出了一倍。但當 IPTV 服務使用者(多人或單人)進入群組時，此協定在傳送的訊息數量、服務提供者的運算量上至少減少 98%；當單一使用者離開群組時，此協定傳送的訊息數量、服務提供者的運算量則相同；當多個使用者同時離開群組時，此協定傳送的訊息數量至少減少 30%、服務提供者的運算量至少減少 40%。最後模擬實驗結果說明本論文所提的樹平衡演算法能夠維持樹平衡。

關鍵字：以頻道為分群基礎的金鑰管理方式、IPTV 服務、金鑰更新演算法、金鑰管理、安全群播、平衡樹、共謀攻擊

A Channel-based Key Management Protocol for IPTV Services

Student: Meng-Ju Lee Advisor: Chi-Chun Lo

Abstract

With the rapid development of network, Internet Protocol Television (IPTV) becomes feasible. IPTV needs a secure, efficient, and scalable key management to prevent unauthorized users watching IPTV's contents. This thesis presents an idea of channel-based key management protocol for IPTV service. In this protocol, IPTV service provider keeps lots of group keys, channel keys, administrative keys, and secret numbers. Each subscriber keeps a group keys and lots of channel keys, administrative keys, and secret numbers. Besides, this thesis proposes rekeying operations including: Join Operation, Leave Operation, Change Operation, and Per-update Operation. This protocol also maintains the efficient rekeying by maintaining balance trees. We analyze our protocol's performance with other related protocols. In security analysis, this protocol more provides forward secrecy, backward secrecy, and collusion attacks prevention. In simulation analysis, three indexes are used: computational costs, number of rekeying messages and storages. Even though, the storages of this protocol are twice bigger. When a member/ members join a group, this protocol is at least 98% better. When a member leaves a group, the service manager's computational costs and number of rekeying messages are same. When members leave a group, numbers of rekeying messages are at least 40% less and the computational costs of service manager are at least 30% less. Finally, the simulation results show that this protocol can keep trees in balance.

Keywords: Channel-based key management, IPTV service, Rekeying operation, Key management, Secure multicast, Tree balance, Collusion attack.

致謝

能夠順利完成這篇論文，首先必須感謝家人無時無刻的支持與鼓勵，讓我在挫折、失落時，能有所依偎與繼續努力的勇氣。此外，也要感謝指導教授羅技群老師，不論在課業或碩論，糾正指導我不足與需要改進的地方，並且在日常生活上對我們付出的關心與照顧，讓我們擁有充實的碩班生涯。

接著尤其要感謝俊傑學長，針對此碩論對我的幫助與指教，既細心且耐心的提醒，對於生活細節也樂於與我們分享討論。也感謝實驗室中的鼎元學長、志華學長、斯寅學長、栩嘉學姊、邦曄學長，讓我了解到做研究的熱誠與精神，熱愛研究的生活，也慢慢地影響著我對研究的態度與做事方式，讓我獲益良多。

最後當然也要感謝實驗室中的學長姊─元辰、世豪、湘婷、志健、致衡、冠儒，以及同學們─秉賢、冠廷、光禹、哲豪、慕均、芳儀、靜蓉、棉媛，能夠在有困難時，不吝嗇地伸出援手，給予實際的幫忙與精神上的鼓勵，分享生活中的點點滴滴，也帶來歡愉的研究氣氛。而研究室中的歡樂氣氛，當然也少不了學弟妹的幫忙─佳蓁、雅晴、雅芬、馨瑩、媛如、御柔、淇奧、彥似、淳皓、漢麟，讓研究生活中仍能保持輕鬆愉悅的心情。

Contents

摘要 I

Abstract II

致謝 III

Contents IV

Lists of Figures VI

Lists of Tables IX

Abbreviations and acronyms X

Chapter 1 Introduction 1

1.1 Backgrounds 1

1.2 Motivations 1

1.3 Objectives 2

1.4 Organization of this Thesis 3

Chapter 2 Related Works 4

2.1 Internet Protocol Television (IPTV) 4

2.1.1 Internet Protocol Television (IPTV) 4

2.1.2 The Features of IPTV 5

2.1.3 The Architecture of IPTV Protocols and IPTV Security Protocols 6

2.1.4 IPTV Transmissions 9

2.2 Key Management Protocols 11

2.2.1 The Security Issues of Multicasting Key Management 11

2.2.2 Key Graphs 12

2.2.3 Classifications of Group Key Management 14

2.3 Sun's et al. 's Conditional Access System 16

2.3.1 The Architecture of F-PPC System 16

2.3.2 Membership Management 18
2.3.3 Extensions of the Trees 19
2.3.4 Improvement of Sun's et al. 's Conditional Access System 20
Chapter 3 A Channel-based Key Management Protocol for IPTV Services 22
3.1 Design Issues 22
3.2 Abbreviations and Acronyms 22
3.3 The Architecture 24
3.4 A Channel-based Key Management Protocol 25
3.4.1 Join Operation 28
3.4.2 Leave Operation 30
3.4.3 Change Operation 33
3.4.4 Per_update Operation 37
3.5 Balance Tree: A Problem of the Proposed Protocol and its solutions 37
3.5.1 Multi-LeaveNode Operation 38
3.5.2 Multi-JoinNode Operation 41
3.6 Discussions 42
Chapter 4: Simulation and Security Analyses 47
4.1 Security Analysis 48
4.2 Simulation Results and Analytical Analysis 49
4.2.1 Simulation Results and Analytical Analysis 49
4.2.2 Simulation Results: Tree Balance 60
4.3 Discussions 63
Chapter 5 Conclusions and Future Works 66
5.1 Conclusions 66
5.2 Future Works 67
References 68

Lists of Figures

Figure1: IPTV infrastructure 6
Figure 2: Detailed IPTV functional architecture 7
Figure 3: IPTV general security architecture 8
Figure 4: IPTV content protection architecture 8
Figure 5: IPTV service protection architecture 9
Figure 6: IPTV services transmissions 10
Figure 7: Multicast 11
Figure 8: User joins or leaves a key tree 13
Figure 9: Two-level trees 17
Figure 10: The R values S4 has known 18
Figure 11: Extension of the tree 19
Figure 12: S2 rejoins the group on the same path related to itself 20
Figure 13: S2 rejoins the group in different leaf node 20
Figure 14: The key structure 25
Figure 15: Subscriber register phase 26
Figure 16: A combined ancestor trees (CAT), when *u3* joins or leaves a group 27
Figure 17: A Sub-CAT, when u3 joins or leaves a group 27
Figure 18: Subscriber joins, leaves, and changes the group 28
Figure 19: Join operation 29
Figure 20: Key structure of Join operation 30
Figure 21: Leave Operation 31
Figure 22: Key structure of Leave operation (1) 32
Figure 23: Key structure of Leave operation (2) 33
Figure 24: A channel group 33

Figure 25: The change operation in channel tree 36
Figure 26: Key structure of change operation 37
Figure 27: Multi-LeaveNode Operation (1) 39
Figure 28: Multi-LeaveNode Operation (2) 40
Figure 29: Multi-JoinNode Operation 42
Figure 30: Key structure of user rejoining 43
Figure 31: Key structure of Leave operation (3) 45
Figure 32: Key structure of Leave operation (4) 46
Figure 33: Number of rekeying messages when a member joins a group 52
Figure 34: The computational costs of service manager when a member joins a group 52
Figure 35: Number of rekeying messages when a member leaves a group 54
Figure 36: The computational costs of service manager when a member leaves a group 54
Figure 37: Number of rekeying messages when members join a group 56
Figure 38: The computation costs of service manager when members join a group 56
Figure 39: Number of rekeying messages when members leave a group 58
Figure 40: The computational costs of service manager when members leave a group 58
Figure 41: Service manager's storages 60
Figure 42: Each member's storages 60
Figure 43: The number of joiners is more than a tree's capacity 61
Figure 44: A tree is 16 leaf nodes with 3 vacant leaf nodes 61
Figure 45: The new tree is 32 leaf nodes with 11 vacant leaf nodes 61
Figure 46: A tree with 32 leaf nodes, and 32 group members (1) 62
Figure 47: A tree with 32 leaf nodes, and 32 group members (2) 62
Figure 48: A new tree after reconstruction (1) 62
Figure 49: A new tree after reconstruction (2) 62
Figure 50: The percentages that IPTVP less than other protocols' number of rekeying

messages in scenario 1 64
Figure 51: The percentages that IPTVP less than other protocols' computational costs in scenario 1 64
Figure 52: The percentages that IPTVP less than other protocols' number of rekeying messages in scenario 3 64
Figure 53: The percentages that IPTVP less than other protocols' computational costs in scenario 3 65
Figure 54: The percentages that IPTVP less than other protocols' number of rekeying messages in scenario 4 65
Figure 55: The percentages that IPTVP less than other protocols' computational costs in scenario 4 65

Lists of Tables

Table 1: The differences among traditional TV, internet TV and IPTV [11] 5

Table 2: User oriented rekey messages 13

Table 3: Key oriented rekey messages 13

Table 4: Group oriented rekey messages 14

Table 5: Abbreviations and acronyms 23

Table 6: Abbreviations and acronyms (2) 47

Table 7: Security analyses with Sun's et al. CAS 48

Table 8: Security analyses with group key protocols 49

Table 9: Efficiency comparisons when a member joins 51

Table 10: Efficiency comparisons when a member leaves 53

Table 11: Efficiency analyses when members join a group 55

Table 12: Efficiency analyses when members leave a group 57

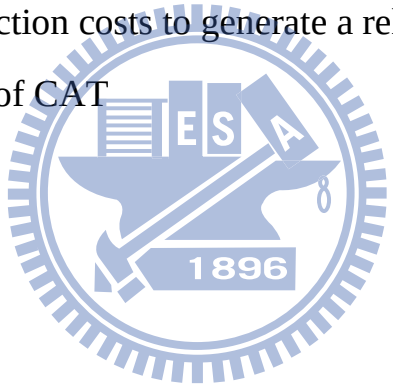
Table 13: Storage analyses 59

Abbreviations and acronyms

RGK_j：	A receiving group key corresponds to group j. Authorized subscribers have the rights to get group messages by using RGK.
AK：	An authorization key or channel key. Authorized subscribers have rights to watch channels by using AK.
MPK_{sk}：	A master private key corresponds to subscriber *k*, s_k. Each subscriber holds it and only the subscriber himself/herself has the key.
KEK_i：	Key encryption key corresponds to node *i*. It is an administration key, and is used to manage key distribution to subgroup members. There are subgroups in a tree, and subgroup members share the same KEK.
R_i：	Relational secrete number corresponds to node *i*.
R_{RGKj}：	Relational secrete number corresponds to root node of group tree and RGK *j*.
RAK：	Relational secrete number corresponds to root node of channel tree and AK.
CAT：	Combined ancestor tree. CAT is those corresponding nodes on the group tree of ancestors and affected leaves, when members join or leave a group.
Sub-CAT：	Subtree of combined ancestor tree. Those corresponding nodes on CAT except root node and its affected leave node.
c：	Total numbers of channels that IPTV service provider provides.
r：	A random number.
s_k：	A subscriber with a serial number *k*.
G_j：	A group with a number *j*.

N_{sk}：	A node corresponds to subscriber *k*.
$KEKs_{Group}$：	A set of KEK in a group tree.
$KEKs_{Channel}$：	A set of KEK in a channel tree.
Rs_{Group}：	A set of R in a group tree.
$Rs_{Channel}$：	A set of R in a channel tree.
AKs：	A set of AK.
KEKs：	A set of KEK
Rs：	A set of R
Rs_{AK}：	A set of R corresponds to root node of channel trees and AKs.
i-1：	Node *i*-1 is the affected leaf node which corresponds to ancestor node *i*.
R_{i-1}：	Node *i*-1 is the affected leaf node which corresponds to ancestor node *i*. R_{i-1} is the affected leaf node's R value.
KEK_{i-1}：	Node *i*-1 is the affected leaf node which corresponds to ancestor node *i*. KEK_{i-1} is the affected leaf node's KEK.
Cs_{Gj}：	A set of channels group *j* subscribes.
Cs_{GjGf}：	A set of channels both group *j* and group *f* subscribe.
CAT_{Gj}：	CAT is those corresponding nodes on the channel tree of ancestors and affected leaves, when G *j* does the join operation or leave operation.
$CAT_{Gj\&Gf}$：	The $CAT_{Gj\&Gf}$ is those nodes which both in CAT_{Gj} and CAT_{Gf}, when CAT_{Gj} and CAT_{Gf} are on a same channel tree.
SKDC：	Simple key distribution center
LKH：	Logical key hierarchy
OFT：	One-way function tree
IPTVP：	IPTV protocol (this thesis's protocol)
CAT：	Combined ancestor tree.

mc.：	Multicast
Num.：	Number
msg.：	message
Comp.：.	computation
n：	Numbers of group members
CE：	Encryption costs
C_r：	The costs of generating a new key from a cryptographically-secure random source
C_f：	Hash function costs
C_K：	Hash function costs to generate a key
C_R：	Hash function costs to generate a relation
sl：	The size of CAT



Chapter 1 Introduction

In this section, there are going to brief describe the backgrounds, motivations, objectives, and the organization of this thesis.

1.1 Backgrounds

In recent years, Internet Protocol Television (IPTV) has become one of the popular multimedia services [8][7][20]. As the rapid development of Internet technology, telecommunication and broadcasting industries will encounter big changes. The diversity devices, such as mobile, personal computer, ipad, set-top-box (STB) , …etc., also bring a new way showing multimedia services. IPTV has been caught attention by people right now, because IPTV is one of the digital convergence technology integrating internet, telecommunication and broadcasting different networks.

Internet Protocol Television (IPTV) means that the digital television was provided over Internet Protocol (IP)[16]. Therefore, IPTV includes lots of services, such as commercial grade multicasting TV, video on demand (VoD), triple play, voice over IP (VoIP), and email. IPTV is far beyond the traditional TV, because it is not only merge data, voice, and video, but also provides more customized products.

1.2 Motivations

IPTV provides more customized and user interactive products which is comparing with traditional pay-television (TV). In the past, traditional TV provides prearranged channel bundles as products to customers in a certain period, and customers only have few products to choose if they want to watch TV channels. Besides, subscribers can not immediately change their subscribing programs. They only can change their program when a new certain period started, and through another communication channel, phone, mail, and going to the shop in person as examples.

Therefore, IPTV provides more fairly and ideal paying programs for customers. IPTV provides users interactive interface, then users could subscribe/unsubscribe or change their services whenever they like. IPTV service provider also provides multiple channels and the protected digital contents to his/her subscribers over wired or wireless network. Hence, IPTV service provider provides customers more flexible products, and allows subscribers choosing their favorite channels on demand.

The key management is one of the security issues for IPTV service. A key management protocol is used to maintain and update the keys provided by the service provider in time. Since the way IPTV transmits data and the infrastructure of IPTV is different with traditional TV and web TV. IPTV transmit digital data through different ways, unicasting, multicasting, per-to-pear for examples and provides the interactive and customized product. Therefore, the traditional key management is not a suitable way to directly implement on IPTV environment. For these reasons, this thesis is focus on designing the key management for IPTV.

1.3 Objectives

This thesis is mainly focus based on the multicasting infrastructure of IPTV, and designs a key management to secure the digital data transmitted through internet protocol. Due to multicasting is the more efficient and feasible way to deliver large real-time data right now. However, there are some security issues for multicast environment, such as confidentiality, authentication, forward secrecy, and backward secrecy. However, collusion attack security should also be attention.

In this thesis, an idea of channel-based key management protocol for IPTV service is proposed to commit those conditions described before. It is a centralized key management system, and has to satisfy security requirements. In some circumstances, collusion prevention is also a must. According to the fluent changing the size of groups and subscribers'

membership changing, there is a rekeying operation used to generate and distribute the new group key for the group. Besides, there are some operations proposed following this thesis to support the rekeying operation, join operation, change operation, and leave operation for example. Analyze and compare different key management will also be in the end of this thesis.

1.4 Organization of this Thesis

Following this thesis, the related paper and research of IPTV or key management are firstly described in the chapter 2, including the IPTV introduction, transmission of IPTV digital data, different kinds of key management, security issues related IPTV key management, and some papers related to this thesis. The design of key management protocol for IPTV service will then be illustrate precisely according to those related works in the chapter 3. In chapter 4, there are analyses and comparisons between the key management of IPTV and the methods other researcher have proposed. In the end of this chapter, the conclusion and future work will be discussed.

Chapter 2 Related Works

Those researches which related with this thesis will mainly be discussed in this chapter. There are four parts in this chapter, including, Internet Protocol Television (IPTV), key management protocols, IPTV key management, and tree nodes arrangement.

2.1 Internet Protocol Television (IPTV)

This part will briefly describe the IPTV, and the ways IPTV transmits digital data, because this thesis is primarily implemented in IPTV environment. Therefore, the introductions of the IPTV and the security issues of the IPTV will be simply described as following.

2.1.1 Internet Protocol Television (IPTV)

IPTV is "multimedia services such as television/ video/ audio/ text/ graphics/ data delivered over IP-based networks managed to provide the required level of QoS/QoE, security, interactivity and reliability" as ITU-T's definition[5]. IPTV is a convergence product of voice, data and video, and could optional be transmitted over next generation network (NGN) [6]. Besides, in the part of transmitting videos, IPTV services could be classified into live television, time-shifted programs and video on demand (VOD) [8]. Therefore, IPTV has its own infrastructure and features. For the feasibility of IPTV, quality of service (QoS) and quality of experiment (QoE) are also concerned.

The IPTV standardizations [11], [4] are defended by many deferent groups. International Telecommunication Union - Telecommunication Standardization Sector (ITU-T, officially CCITT) is one of the groups, and they mainly focus on defining the globalize IPTV standardization. ITU-T also cooperates with other groups to define IPTV, like Alliance for Telecommunications Industry Solutions - IPTV Interoperability Forum (ATISIIF)[1],and Digital Video Broadcasting (DVB)[2] as examples. However, another group called Open IPTV Forum [13] defines its own IPTV specifications. There

are some groups working on the different areas standards, and those standards will also affect the way IPTV implementation. Such as European Telecommunications Standards Institute (ETSI) also called Telecoms & Internet converged Services & Protocols for Advanced Networks (TISPAN)[18], is working on the next generation network (NGN) which IPTV services will be implemented on.

2.1.2 The Features of IPTV

Selection, storage, QoS, and low cost are the features of IPTV [16]. Besides, customization and the interactive feature are the main factors that differentiating the traditional TV and IPTV. Internet TV and IPTV are also have differences. Table 1 shows the differences among IPTV and traditional TV and internet TV in detail.

Table 1: The differences among traditional TV, internet TV and IPTV [11]

Traditional TV (Terrestrial broadcasting TV, cable TV, Satellite TV)	Internet TV	IPTV
Limited interactivity	Full interactivity	Full interactivity
Broadcast all channels all the time	Broadcast only channels being watched at given time	Broadcast only channels being watched at given time
Limited content	Unlimited content	Limited content (which depends on the service providers)
Set-top-box with a television display	PC	Set-top-box with a television display
Low level viewer involvement	High level viewer involvement	Low level viewer involvement

Services not guaranteed	Best effort quality, QoS not guaranteed	Guaranteed QoS
Users and media data are protected	Unsafe	Users and media data are protected
Known customers and locations	Any user (generally unknown)	Known customers and locations by IP address

2.1.3 The Architecture of IPTV Protocols and IPTV Security Protocols

There are four main complements consist IPTV [6], including contents providers, service providers, network providers, and customers, as Figure1: IPTV infrastructure described. And each component has different devices to receive and transmit IPTV data packages.

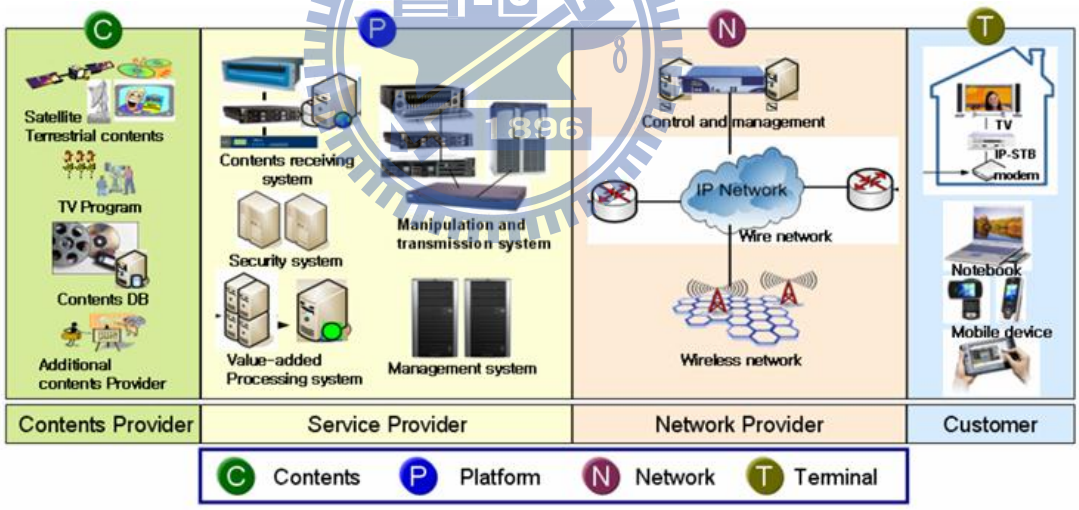
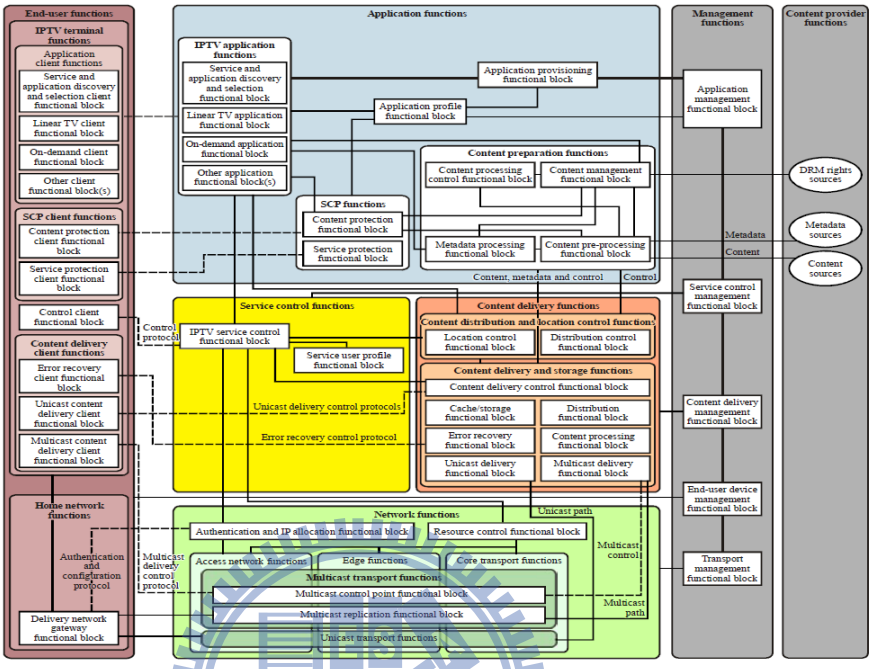


Figure1: IPTV infrastructure

Source: [20]

ITU-T also defines the logical functional infrastructure of IPTV in Y.1910, Figure 2. Those logical functions in the architecture are working to control and make IPTV work, like end-user functions, application functions, service control functions, content delivery functions, network functions, management functions, and provider

functions.The functions related to IPTV security is service and content protection (SCP) functions which not only residing in application functions and end-user functions.



NOTE 1 – For NGN IPTV architectures, "authentication and IP allocation functional block" is replaced by "network attachment control functions (NACF)".
NOTE 2 – For NGN IPTV architectures, "resource control functional block" is replaced by "resource and admission control functions (RACF)".
NOTE 3 – For NGN IMS IPTV architecture, "IPTV service control functional block" is replaced by "core IMS functions".
NOTE 4 – For NGN IMS IPTV architecture, "control client functional block" is replaced by "session client functional block".

Figure 2: Detailed IPTV functional architecture

Source: [6]

The logical function architecture of IPTV security delineate in X.1191, Figure 3. The security specifications of IPTV are divided to two areas. However, one the area is not the primary topic in X.1191 due to private agreements, and the content provider domain and the interconnection between content providers and service providers are involved in one of the area [3]. The SCP functions are reside primarily to content protect functions and service protect functions. Content protect functions are related to rights, key management content encryption…etc., as Figure 4. Instead, the service protect functions are related to authentications, authorizations and access controls…etc., as Figure 5.

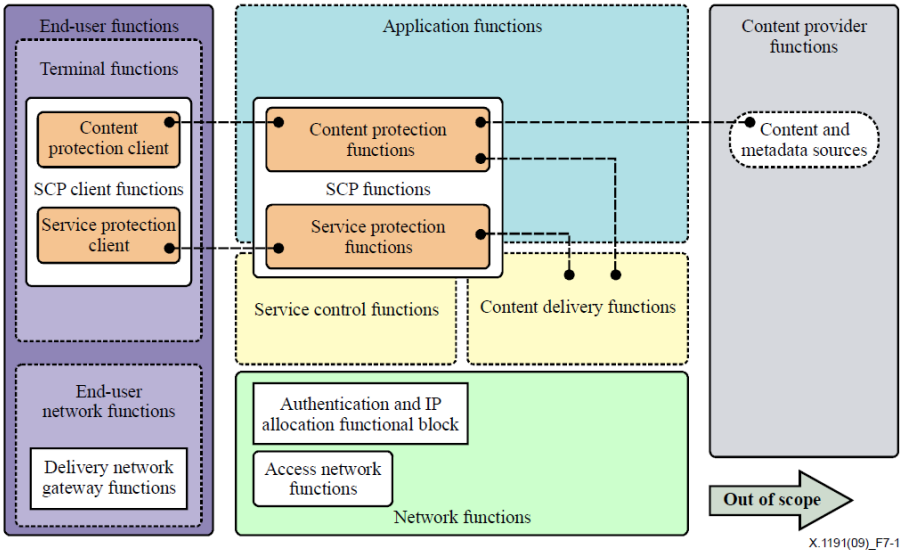
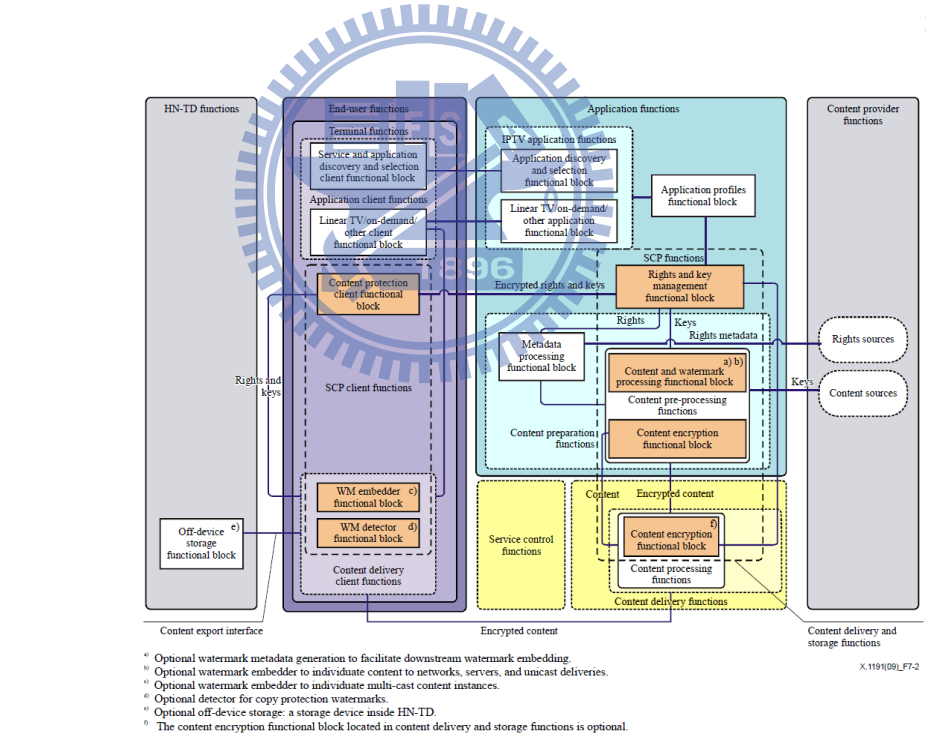


Figure 3: IPTV general security architecture

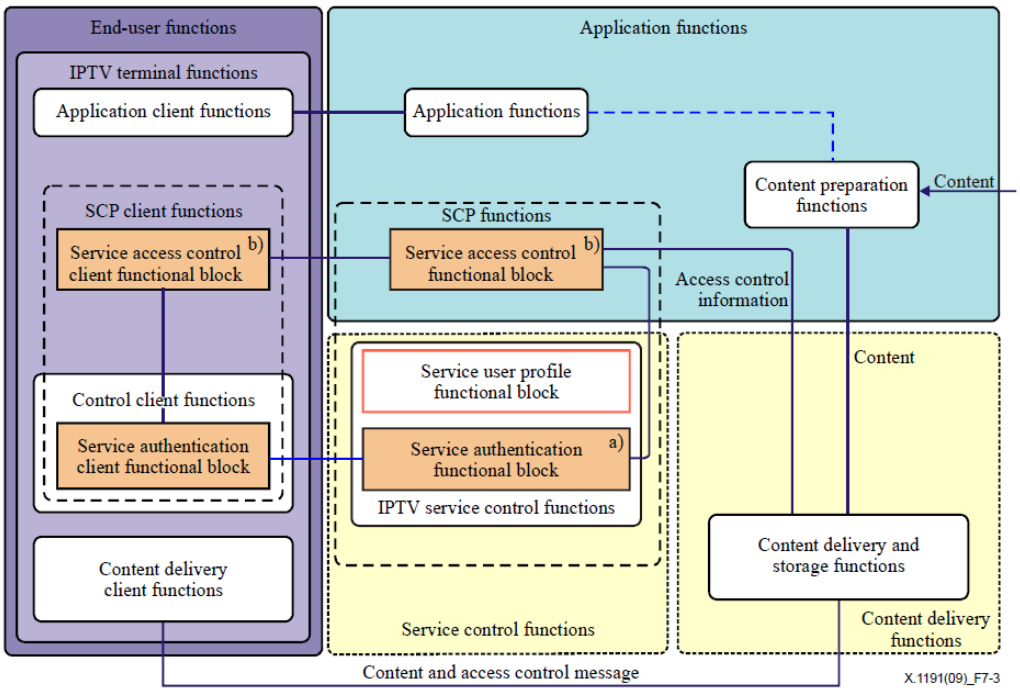
Source: [3]



NOTE – The content protection functional blocks in this figure consist of content protection functions and content protection client functions.

Figure 4: IPTV content protection architecture

Source: [3]



[a] Authentication: It identifies a subscriber name and ID with the assigned privilege.
[b] Service access control: To protect a service from the illegally unauthorized access.

NOTE – The service protection functional blocks in this figure consist of service protection functions and service protection client functions.

Figure 5: IPTV service protection architecture

Source: [3]

2.1.4 IPTV Transmissions

IPTV services transmission parties could be divided into IPTV contents, core network, access network and IPTV user devices [16], Figure 6. Unicast (or Point-to-Point), multicast and pear-to-pear (P2P), are three methods to transmit IPTV services. Unicast is high interactive way connecting services and subscribers, but its only can serve small group of users. P2P is different from the service and client architecture; instead each user could be server and client. P2P is a great solution for a scale of users, but P2P also brings users to a seriously dangerous situation due to the unknown and not secure resources.

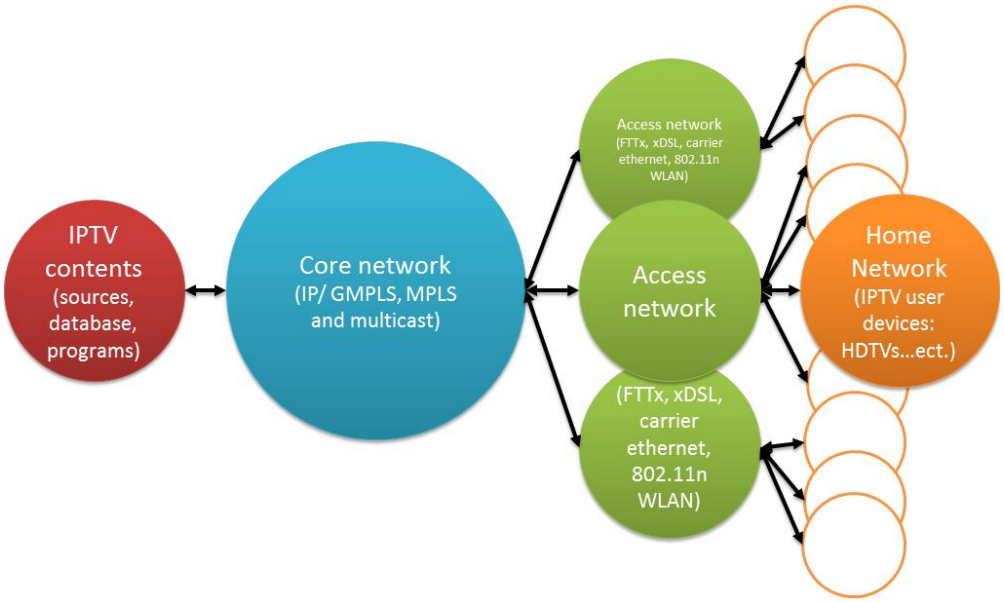


Figure 6: IPTV services transmissions

Multicast could deliver data to a specific group through IGMP, GMPLS, MPLS…etc., as Figure 7. Multicast transmits data through class D where IP address range from 224.x.x.x to 239.x.x.x, IGMP as an example. Dense multicast, sparse multicast, and source-specific multicast are the different methods for multicast IPTV services [16]. Multicast is the most ideal way to transfer IPTV services. Even though, multicast has scalability problem, it still catch people's attention. Multicasting could save bandwidth of core network and access network if there is more than one user watching the same programs. In this thesis, the key management is focusing on the multicast and unicast transmissions instead of P2P protocol.

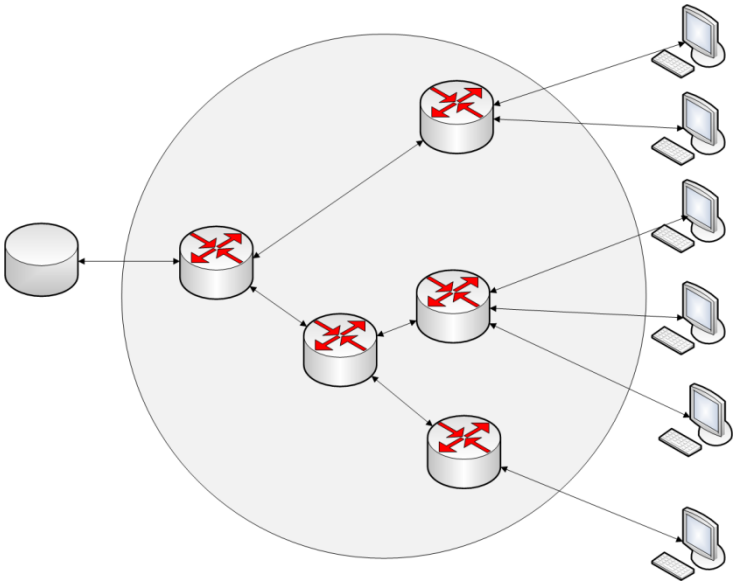


Figure 7: Multicast

2.2 Key Management Protocols

This chapter is mainly focusing on the group key management in multicasting IP network. Hence, this chapter will introduce the security issues, the different key management, and key graphs for group communication.

2.2.1 The Security Issues of Multicasting Key Management

There are three ways to transmit IPTV services through IP network as described before. The main purpose of secure multicasting is not only protecting data's confidentiality, integrity, and authentication, but also other security issues as following [12]:

- Only group members can get the content of group message.
- The source has been authenticated before being transmitted to group members.
- New group members could not get the content of group data transmitted before they join the group (Forward secrecy).
- Leaving members could not get the content of group data after they are

revoked from the group (Backward secrecy).

A key management system has to satisfy four security requirements: confidentiality, authenticity, backward secrecy, and forward secrecy. Collusion prevention is also a must in some circumstances.

According to those security issues mentioned before, the group key is needed. Group key is an efficient way to securely manage group communications through multicasting IP network. Therefore this thesis proposed a group key management for the IPTV services.

2.2.2 Key Graphs

There is a scalability problem of group/multicast key management. The problem becomes more seriously when a member joins or leaves the group. Then, the server need to rekey, updating the group key, to make sure the forward secrecy and backward secrecy. There are three rekeying strategies were discussed as following [19]:

- User oriented rekeying

  Sending rekeying message to each user by using the keys user has. For example, when u4 is going to join/leave the secure group in Figure 8, server generates the rekey messages and sends to the group members, as shown in Table 2.

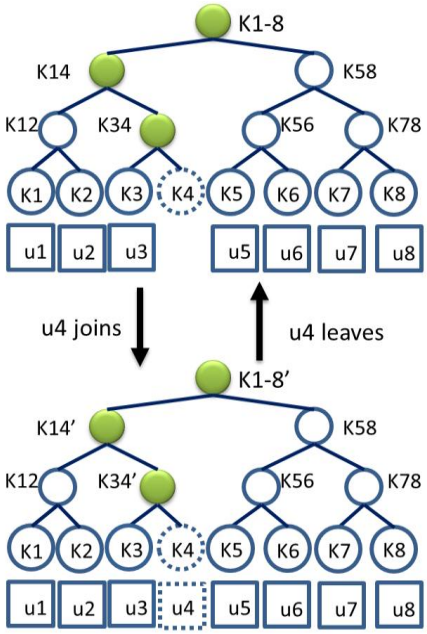


Figure 8: User joins or leaves a key tree

Table 2: User oriented rekey messages

u4 joins the key tree	u4 leaves the key tree
s->{u4,…, u8}: {k1-8'}k1-8	s->{u4,…, u8}: {k1-8}k58
s->{u1, u2}: {k1-8', k14'}k14	s->{u1, u2}: {k1-8, k14}k12
s->{u3}: {k1-8', k14', k34'}k34	s->{u3}: {k1-8, k14, k34}k3
s->{u4}: {k1-8', k14', k34'}k4	

- Key oriented rekeying

  Server sends new keys encrypted by old keys to group members, when user joins the group. Instead, server sends new keys individually encrypted by unaffected old keys. An example shows in Table 3.

Table 3: Key oriented rekey messages

u4 joins the key tree	u4 leaves the key tree
s->{u1,…, u8}: {k1-8'}k1-8	s->{u4,…, u8}: {k1-8}k58
s->{u1, u2, u3}: {k14'}k14	s->{u1, u2}: {k1-8}k14, {k14}k12

s->{u3}: { k34'}k34 s->{u4}: {k1-8', k14', k34'}k4	s->{u3}: {k1-8}k14, {k14}k34, {k34}k3

- Group oriented rekeying

  Server sends only two messages to the group when a user joins the group. One message contains all the new keys and broadcasts to the whole group members, and the other one message only transmits to the join member. Server only needs to send one message to group members when a user leaves the group, instead. Table 4 shows an example in Figure 8.

Table 4: Group oriented rekey messages

u4 joins the key tree	u4 leaves the key tree
s->{u1,…, u8}: {k1-8'}k1-8, {k14'}k14, { k34'}k34 s->{u4}: {k1-8', k14', k34'}k4	s->{u1,…, u8}: {k1-8}k58, {k1-8}k14, {k14}k12, {k14}k34, {k34}k3

For reducing the loading of server, it is better to take group oriented rekeying strategy then the key oriented and user oriented strategies. On the other hand, customer's processing loading will increase gradually as taking user oriented, key oriented, and then the group oriented rekeying strategies. According to the recent network transmission ability, bandwidth, and user's computing power limitation…etc., server managers would have a different choice of the rekeying strategies.

2.2.3 Classifications of Group Key Management

In generally, Group key management protocols could reside primarily in three categories: centralized, decentralized, and distributed. The simplest way to manage

group members has scalable limitation, however. Such as Simple Key Distribution Center (SKDC), a group manager using different keys and each key corresponds to a group member. If a group with n members, there are n rekeying messages and n encryptions when a member join or leave a group. Therefore, the more scalable way are as following [14]:

- Centralized group key management protocols

  In centralized protocols, a key server (KS) is employed for controlling group activities. Most of them, such as LKH, OFT [15], etc., are tree-based. A tree-based approach employs a hierarchy of keys in which each group member, based on his/her location, and is assigned a set of keys. For a group of n n members in a $k-$ary tree-based group key management system, each member keeps $\log_k n$ administration keys and KS has to send $(k-1) * \log_k n$ rekeying messages. Therefore, minimizing the storage, computational power and bandwidth utilization on both client and server are critical issues. Numbers of packages sent by server also need to be concerned.

- Decentralized group key management protocols

  In decentralized protocols, a network is divided into many subgroups. Each subgroup has a group manager to manage their own subgroups. The advantage is that if a subgroup manager goes down, it won't affect whole system. However, there is still a main system manager connecting all subgroup managers. When the system manager stops working, the whole system is still affected, and subgroups could not communicate with each other.

- Distributed group key management protocols

  In distributed protocols, there is no group manager, and the group key is generated and maintained by the group members themselves. The group key is usually generated together by all members contributing their own secure

information due to security requirements and each member's computation power. When the members grow up, the processing and communication time are also linearly increasing. To maintain the protocols robust, group membership list needs to be take care too.

In this thesis, we focus our paper on the centralized ones due to the flexibility of payments as mentioned in the motivation. The channel-based group key management, this thesis proposed, revises from the Sun's et al. 's conditional access system[17]. Following sections will delineate the method.

2.3 Sun's et al. 's Conditional Access System

There is a new model of conditional access system (CAS) proposed by Sun's et al [17]. The model is different from traditional pay-TV broadcasting system. Instead it provides scalable flexible-pay-per-channel (F-PPC) system, and all subscribers could change their subscriptions any time. The architecture of F-PPC system and membership management will be described in order.

2.3.1 The Architecture of F-PPC System

In F-PPC system, it implements four-level key hierarchy, and brings the key updating processes more efficiency. Those four-level keys are:

- Control word (CW): a random number generated by server
- Authorization key (AK): Authorize subscribers rights to watch channels
- Receiving group key (RGK): Authorize subscribers the rights to get group messages.
- Master Private Key (MPK): Each subscriber holds it and only the subscriber himself/herself has the key.

This new CAS system uses tree architectures to manage keys. There are two-level trees in F-PPC system. One is group trees and the other one is channel trees. F-PPC groups

subscribers and construct group trees according to their subscriptions and their selected channels, as shown in Figure 9. There are at most $1 \le N \le \min(2^C, S)$ groups, which the N denotes group numbers, C denotes the channels and S denotes the numbers of subscribers. Then the subscribers in the same group subscribe same channels and share the same group key.

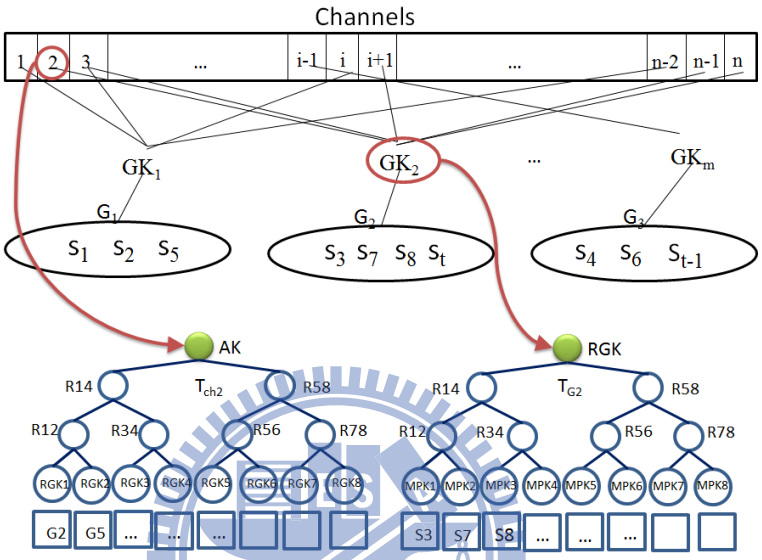


Figure 9: Two-level trees

If a user in one of the group tree, he/she will have MPK, AKs, RGK, and Rs. There are Rs as secret numbers corresponding to each trees' nodes in this F-PPC system. Each user knows the all R values of the group tree instead of the path nodes related to the user. When a user leaves the group, the other group members updates their group keys by exclusive the old key with the R related to the leaving user, such as $RGK' = RGK \oplus R_{leaving\_user}$. Users do not remember all the values of R, and users generate Rs by using left or right hash function.

For example, as shown in Figure 10, S4 knows all the R values instead of R1-8, R14, R34 and R3. According to the left and right function, S4 only needs to remember the values of R12, R3, and R58. The other R values could be calculated, such as R6=HR (HL (R58)). When S4 leaves the group, the server broadcasts a message {LEAVE, S4}, and group members will updating group key (RGK) by $RGK' = RGK \oplus R4$.

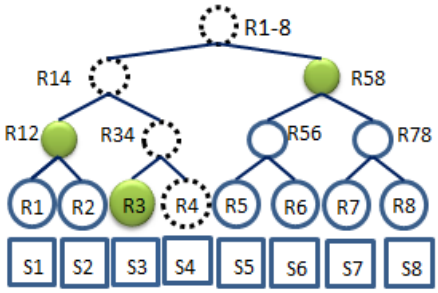


Figure 10: The R values S4 has known

2.3.2 Membership Management

There are five phases in this new CAS model:

- Initial phase: The server constructs two-level trees and generates the AK keys for channel trees and RGK keys for group trees.
- User registration phase: When a user joins this system, service provider will assign him /her a unique identity (ID) and a MPK through a secure channel.
- Subscribing phase: When a user subscribes channels, server will classify him/her to a group. Then server broadcasts join message to the group members, and the group members could generate the R secrete number related the joining member. Thirdly, server transmits RGKs and R_{RGK} encrypted by user's MPK, and also transmits AKs and R_{AK} encrypted by RGKs.
- RGK updating phase: When a group member leaves, the server needs to update RGK key. The server first broadcasts a {LEAVE, node corresponding to leaving-member}, then the rest members in the group will update their RGK key by $RGK' = RGK \oplus R_{leaving\_user}$
- AK updating phase: When a user changes the subscription of certain channels, the channel keys (AKs) must update. For example, when u_k leaves G_j, server firstly broadcasts the leave message {LEAVE, G_j, Nu_k}, then the AK keys is updated by $AK' = AK \oplus R_{Gj.}$ The R_{Gj} is corresponding to the

secret number of G_j in channel trees which G_j is subscribed.

2.3.3 Extensions of the Trees

When the tree is full, the tree needs an extension of one more level from the old tree. The old trees' leaf nodes are move to the left child nodes in the new tree, as shown in Figure 11. Server needs to transmit the new sibling node's R values to the group member, after the extension of the tree and new members join the group.

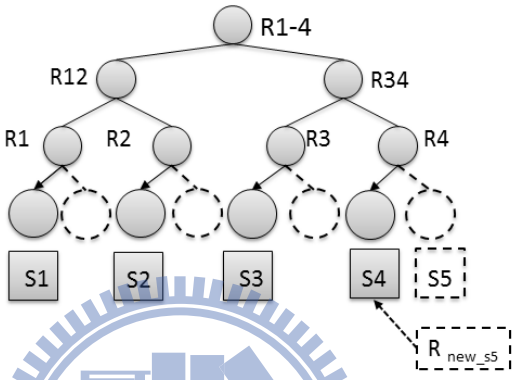


Figure 11: Extension of the tree

Besides, server assigns the old member in the same leaf nodes, when an old member rejoins the same group again. The reason is that each member should not know the R value on the path related to them. If the origin old leaf node is occupied, there are two ways to handle this situation. Firstly, server would extend the origin old leaf node one more level, and move the origin old leaf node to be left child node of new level, Figure 12. The extension to the tree will make un-balance tree, and rearrange the tree periodically would be necessary. Secondly, server just place old member to the other vacant leaf node and update the information old member known before, Figure 13 as an example.

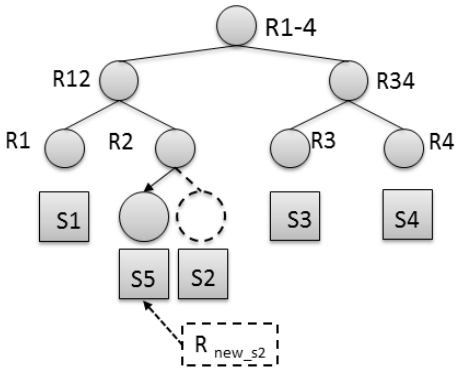


Figure 12: S2 rejoins the group on the same path related to itself

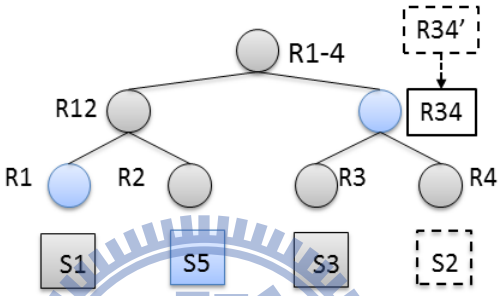


Figure 13: S2 rejoins the group in different leaf node

2.3.4 Improvement of Sun's et al. 's Conditional Access System

Jung-Yoon Kim and Hyoung-Kee Choi [9] proposed two weaknesses and solve those problems. Firstly, the key updating algorithm was so simple that a member could get R value related to its self, when the member rejoin same group and is at the same leaf node. For example, when a subscriber leaves the group, the group key is updated as RGK'=RGK$\oplus$R$_S$. However, when a subscriber rejoins the group, he/she will get the new RGK'. The new RGK' could easily calculate the R value by Rs= RGK'$\oplus$ RGK. The other weakness of the F-PPC system is forward secrecy problem, and group keys and channel keys were not updated when a member joins a group

As those reason mentioned, the improvement of Sun's et al. 's CAS is the robust algorithm and to satisfy both backward and forward secrecy. The keys need to be updated whenever a user joins or leaves the group. For example, if a user joins the group, the RGK is updated by RGK'=RGK$\oplus$R$_S$ which is same algorithm with Sun's et al.

CAS. RGK is updated by RGK'=H(RGK$\oplus$R$_S$) when a group member leaves instead.

This thesis will implement the architecture of Sun's et al. 's CAS system on IPTV services. The both forward and backward securities also take into concerned. In order to avoid collusion attack and solve rejoining members' problem, there is another way to manage the membership in those groups in next chapter.

Chapter 3 A Channel-based Key Management Protocol for IPTV Services

In this chapter, an idea of channel-based key management protocol for IPTV service is proposed. The motivations and objectives of this thesis are mentioned in chapter one. In chapter two, the IPTV services, related security issues, and key management protocols explain the reason of choosing centralized key structure and group key management, and also illustrate problems in Sun's et al. 's CAS. Following this chapter, there will define the problems first, and then delineate the channel-based key management protocol for IPTV services.

3.1 Design Issues

This thesis is going to solve the critical issues based on the Sun's et al.'s CAS:

- Collusion attack prevention: Leaving users could not get the keys and contents from group messages by working together in this thesis. It is also impossible to get the keys and contents from group messages, when leaving members leave group one after another.
- Refreshments of keys and R values: There are forward and backward secrecy problems in Sun's et al.'s CAS proposed and solved by [9]. Nevertheless, there is another way to achieve both forward and backward secrecy by updating keys and R values proposed in this thesis, when members join/leave/change the group.
- Tree rebalances: The trees easily become unbalanced trees, when group members join/leave groups. Reconstructions and maintenances keys in the tree also take in to concern in this thesis.

3.2 Abbreviations and Acronyms

There are lots of abbreviations and acronyms in this thesis. Table 5 shows the

explanations of those abbreviations.

Table 5: Abbreviations and acronyms

Index	Explanations
RGK_j	A receiving group key corresponds to group *j*. Authorized subscribers have the rights to get group messages by using RGK.
AK	An authorization key or channel key. Authorized subscribers have rights to watch channels by using AK.
MPK_{sk}	A master private key corresponds to subscriber *k*, s_k. Each subscriber holds it and only the subscriber himself/herself has the key.
KEK_i	Key encryption key corresponds to node *i*. It is an administration key, and is used to manage key distribution to subgroup members. There are subgroups in a tree, and subgroup members share the same KEK.
R_i	Relational secrete number corresponds to node *i*.
R_{RGKj}	Relational secrete number corresponds to root node of group tree and RGK *j*.
R_{AK}	Relational secrete number corresponds to root node of channel tree and AK.
CAT	Combined ancestor tree. CAT is those corresponding nodes on the group tree of ancestors and affected leaves, when members join or leave a group.
Sub-CAT	Subtree of combined ancestor tree. Those corresponding nodes on CAT except root node and its affected leave node.
c	Total numbers of channels that IPTV service provider provides.
r	A random number.
s_k	A subscriber with a serial number *k*.
G_j	A group with a number *j*.
Ns_k	A node corresponds to subscriber *k*.
$KEKs_{Group}$	A set of KEK in a group tree.

$KEKs_{Channel}$	A set of KEK in a channel tree.
Rs_{Group}	A set of R in a group tree.
$Rs_{Channel}$	A set of R in a channel tree.
AKs	A set of AK.
KEKs	A set of KEK
Rs	A set of R
Rs_{AK}	A set of R corresponds to root node of channel trees and AKs.
$_{i-1}$	Node *i*-1 is the affected leaf node which corresponds to ancestor node *i*.
R_{i-1}	Node *i*-1 is the affected leaf node which corresponds to ancestor node *i*. R_{i-1} is the affected leaf node's R value.
KEK_{i-1}	Node *i*-1 is the affected leaf node which corresponds to ancestor node *i*. KEK_{i-1} is the affected leaf node's KEK.
Cs_{Gj}	A set of channels group *j* subscribes.
Cs_{GjGf}	A set of channels both group *j* and group *f* subscribe.
CAT_{Gj}	CAT is those corresponding nodes on the channel tree of ancestors and affected leaves, when G *j* does the join operation or leave operation.
$CAT_{Gj\&Gf}$	The $CAT_{Gj\&Gf}$ is those nodes which both in CAT_{Gj} and CAT_{Gf}, when CAT_{Gj} and CAT_{Gf} are on a same channel tree.

3.3 The Architecture

There four-level key hierarchy and two-level trees constructed in this protocol which same with Sun's et al. 's CAS, Figure 9. IPTV services are the convergence of telecommunication, internet protocols (IP), and broadcasting network. It is more suitable implement four-level key hierarchy structure, so this architecture could both feasible on both the broadcasting and IP network.

The centralized tree structure and group key managements are implemented. In this

thesis, we assume that there are *n* channels which are provided by the IPTV service provider. Each possible combination of these channels is assigned a group key to protect these programs. Therefore, there are, at most, 2^c*-1* group keys which the IPTV service provider must kept. *c* are total numbers of channels IPTV service provider provides. Moreover, the subscribers are classified into groups. A group will share the same group key, and the group keys are mutually different from each other. As shown in Figure 9, the subscribers in the same group subscribe same channels and share the same group key.

In other words, each user needs to keep MPK, RGK, AKs, R_{RGK}, and R_{AK} in Sun's et al. 's CAS. In this thesis, each user needs to keep additional key encryption keys (KEKs). Besides, R values are generated by service provider using KEK and a random number under a hash function, R_i=H (KEK_i, r). KEKs are used to manage key distribution to subgroups, so server could avoid collusion attacks. The key structure shows in Figure 14.

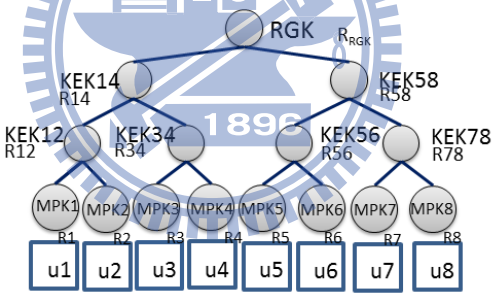


Figure 14: The key structure

3.4 A Channel-based Key Management Protocol

In this section, an idea of channel-based key management protocol for IPTV service is proposed. To prevent an illegal access to IPTV services, there are three phases in the paper. They are:

- Subscriber register phase:

  Users register to IPTV service provider to get login ID and password. After that, they become the member of the IPTV service, and get authority to login to the system by sending Account ID and password to the center server. On the other hand, users gets

a MPK and a random number through a secure channel, and generate R_{user} by themselves through hash function, $R_{user}=H(H(ID_{user}),MPK_{user} \oplus r)$ as an example, Figure 15.

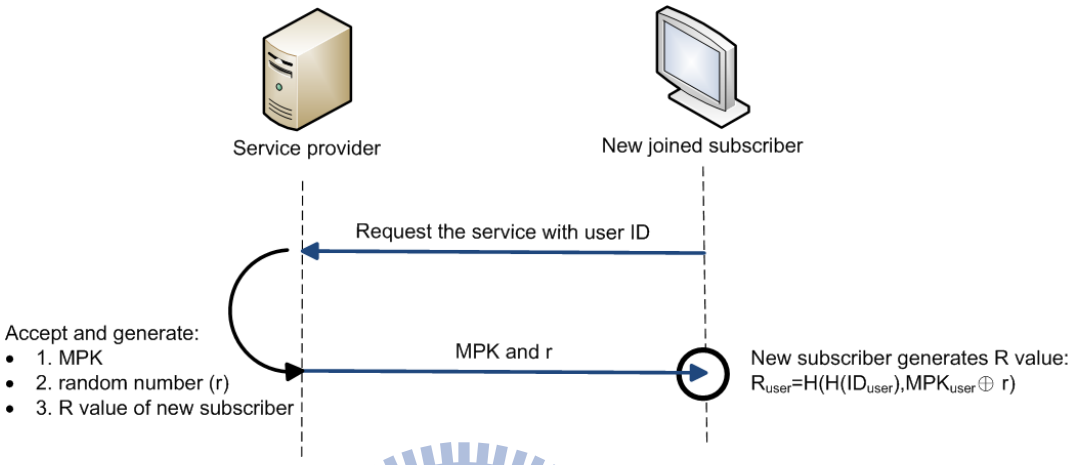


Figure 15: Subscriber register phase

- Channel subscribing phase:

Users subscribe lots of channels. According to their subscribing information, the IPTV service provider gathers the subscribers into a group and they subscribe the same channels. Each group is assigned a group key used to encrypt and protect these programs. The group key is kept by each subscriber, Figure 9. In addition, each subscriber keeps some administration keys used to encrypt the group key, Figure 14. Hence, in the proposed paper, each subscriber has to keep four types of keys: receiving group key (RGK), channel key (AK, authorization key), administration key (KEK, key encryption key), and master private key (MPK).

- Membership management phase:

Membership management is used to manage those possible activities about a subscriber. There are some scenarios: a new subscriber joins a group; an old subscriber cancels his/her subscription; an old subscriber changes his/her subscription. Any kinds of scenarios in the above, the group key has to be refreshed and then distributed to the

group members by rekeying operations.

There are rekeying operations: join, multi-join, leave, multi-leave and change operations. Those rekeying operations work on the combined ancestor trees (CAT) [15] or Subtree of combined ancestor tree (Sub-CAT). CAT is those corresponding nodes on a group tree of ancestors and affected leaves, when members join or leave a group, as shown in Figure 16. When u3 joins a group, the CAT is those gray nodes and the size of CAT are numbers of gray nodes. Sub-CAT is those corresponding nodes in CAT except root node and it's affected leave node. As shown in Figure 17, when u3 joins/leaves a group, the Sub-CAT is those gray nodes. The size of Sub-CAT are numbers of gray nodes in Figure 17. Hence, the last phase is used to manage the generating and distributing of the group keys by rekeying operations.

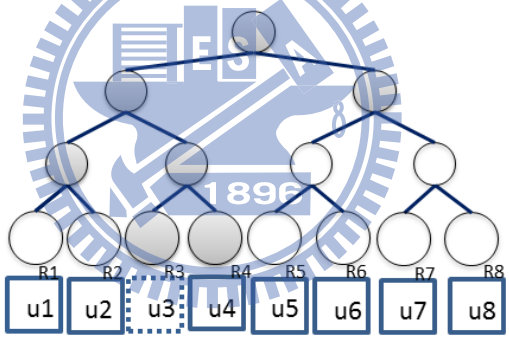


Figure 16: A combined ancestor trees (CAT), when *u3* joins or leaves a group

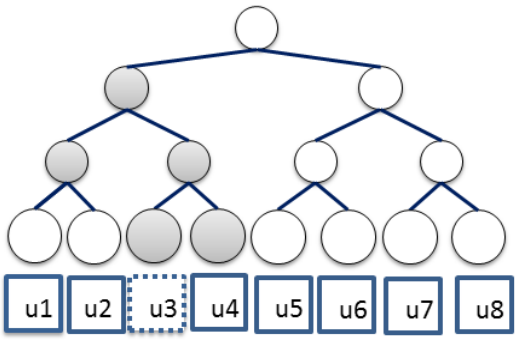


Figure 17: A Sub-CAT, when u3 joins or leaves a group

When a group membership is changed, a rekeying operation is used to generate and distribute the new group key for the group. And Per_update Operation is used to execute a

periodical update when a membership is not changed. In addition, a subscriber will not be in more than one group.

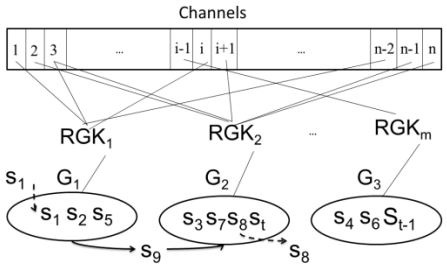


Figure 18: Subscriber joins, leaves, and changes the group

In Figure 18, a new user, s_1, registers to the IPTV service and subscribes lots of channels. According to the subscription information about s_1, he/she will be assigned to a group, G_1, and get the new group key GK_1' and some relation numbers (R values) and administration keys (KEKs).

Because of the paper limitation, this thesis focuses on depicting the concept of the last phase. The following are the operations details.

3.4.1 Join Operation

The join operation is used when a new subscriber, s_k, joins the group. Server broadcasts a message. After group members receive the message, they will automatically updated group key by using R value and hash function. There are some group members who relate to the path from leaf s_k to the root. They also can automatically update key encryption keys and R values (KEK_i', R_i'). There is an example as following, Figure 19:

There is a procedure when a subscriber (s_k) joins group j (G_j):

1. Server broadcasts {JOIN, G_j, Ns_k}
2. Existing group members can automatically updates the ancestors in CAT of s_k. Members, who relate to those ancestors in CAT, will update RGK, KEK_i, R_i and R_{RGK}.

   1) RGK_j and all KEKs update by $RGK_j'=H(RGK_j,R_{RGKj})$,

KEK_i'=H(KEK_i,R_i).

2) The R_i and R_{RGK} value are updated by R_i'=H(KEK_i',R_i) and R_{RGKj}'=H(RGK_j',R_{RGKj})

3. Server unicasts to s_k: MPK_{sk} {RGK_j',$KEKs_{Group}$', R_{RGKj}', Rs_{Group}', AKs', $KEKs_{Channel}$' , Rs_{AK}', $Rs_{Channel}$'}

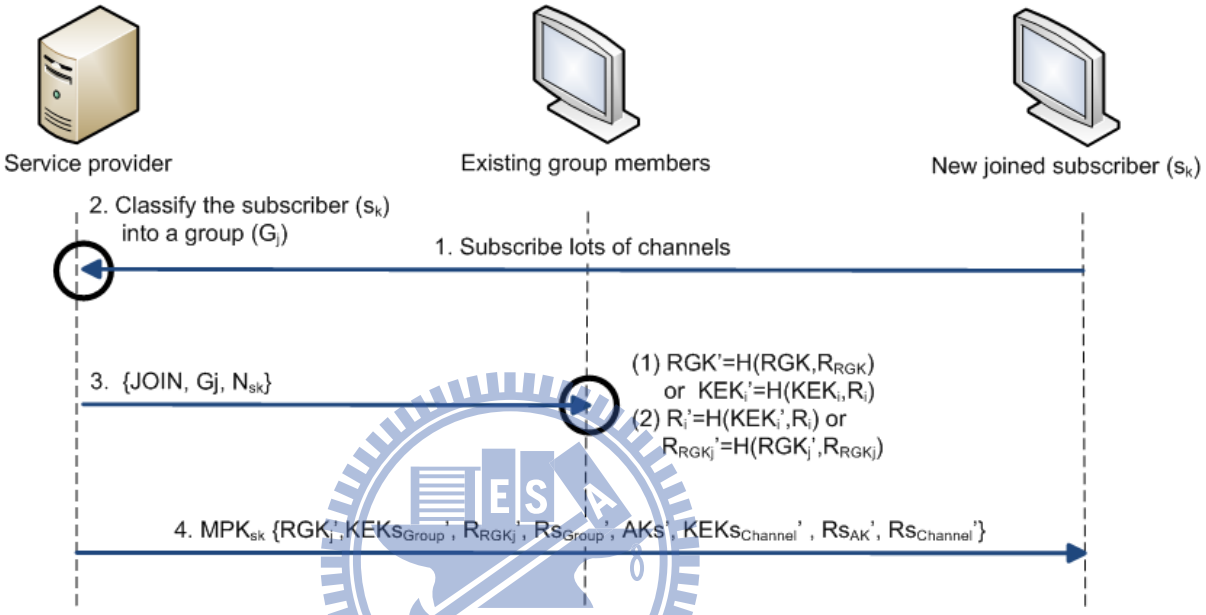


Figure 19: Join operation

More specifically, as shown in Figure 20, there is a new joiner (u3). The IPTV service provider broadcasts a message {JOIN, G_1, N_{u3}} to all subscribers in Operation Join. The group key (RGK_1') and R value (R_{RGK1}') are updated by all group members, RGK_1'=H (RGK_1, R_{RGK1}), R_{RGK1}'=H (RGK_1', R_{GK}). Key encryption key (KEK_{14}) and R value (R_{14}) which u1 and u2 hold are automatically updated by himself/herself, KEK_{14}'=H (KEK_{14}, R_{14}) and R_{14}'=H (KEK_{14}', R_{14}). Key encryption key key (KEK_{34}) and R value (R_{34}) of u4 are updated by himself/herself using XOR operation under a hash function, KEK_{34}'=H(KEK_{34},R_{34}) and R_{34}'=H (KEK_{34}', R_{34}). Then service provider unicasts a message, {RGK_1', KEK_{34}',KEK_{14}', R $_{RGK1}$', R_{34}',R_{14}', AKs', $KEKs_{Channel}$' , Rs_{AK}', $Rs_{Channel}$'}, encrypted by MPK_{u3}.

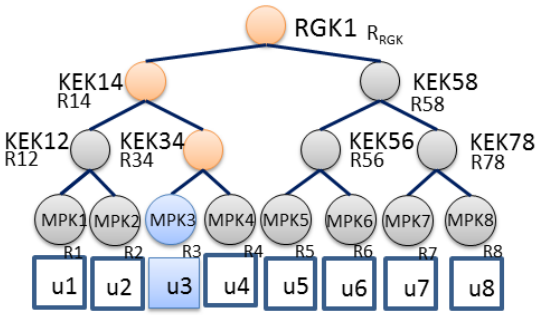


Figure 20: Key structure of Join operation

For efficiency, server joins multiple users in a group at a time. Multi-Join Operation is triggered in that circumstance. The Multi-Join Operation processes are same with Join Operation. However, service provider generates new KEK' and R' values and directly transmits those to those leaf nodes, if an interior node which it's leaf node are both new joined members. Take Figure 20 as an example. If u3 and u4 both are new joiner, service provider will generate new KEK_{34}', and R_{34}'. R_{34}' is generated by using R_{34}'=H (KEK_{34}', r). Then, service provider unicasts messages to u3 and u4.

In this way, IPTV service provider could prevent the new subscriber get data before he/she joins the group and provide forward secrecy.

3.4.2 Leave Operation

Leave Operation is used when an old subscriber, (s_k), leaves the group (G_j). Server will broadcasts a leave message to the entire member in the service. Group members will automatically update key encryption keys and R values, if he/she relates to the Sub-CAT. After that, service provider generates and transmits new group key to the rest. New key encryption keys (KEKs') are transmitted to those members who held it. R values also are updated by members themselves, when those corresponding nodes key encryption keys are changed. An example is as following, Figure 21:

1. Server broadcasts {LEAVE, G_j, Ns_k,}
2. Users who relate to Sub-CAT automatically update ancestors of Sub-CAT by using affected leafs' R value of Sub-CAT. Those updated ancestors include

KEKs, Rs. Node *i*-1 is the affected leaf node which corresponds to ancestor node *i*. R_{i-1} is the affected leaf node's R value.

- KEK_i' updates by KEK_i''=H(KEK_i', R_{i-1})

3. Server generate new RGK''
4. Server transmits new RGK'' and KEK'' to those user who need to know. Those new RGK'' and KEK'' are transmitted to user and encrypted by new KEK'', KEK'' {RGK'', KEK_{i-1}'',…}.
5. Each user automatically updates R_i' , R_i' updates by R_i''=H(KEK_i'',R_i').

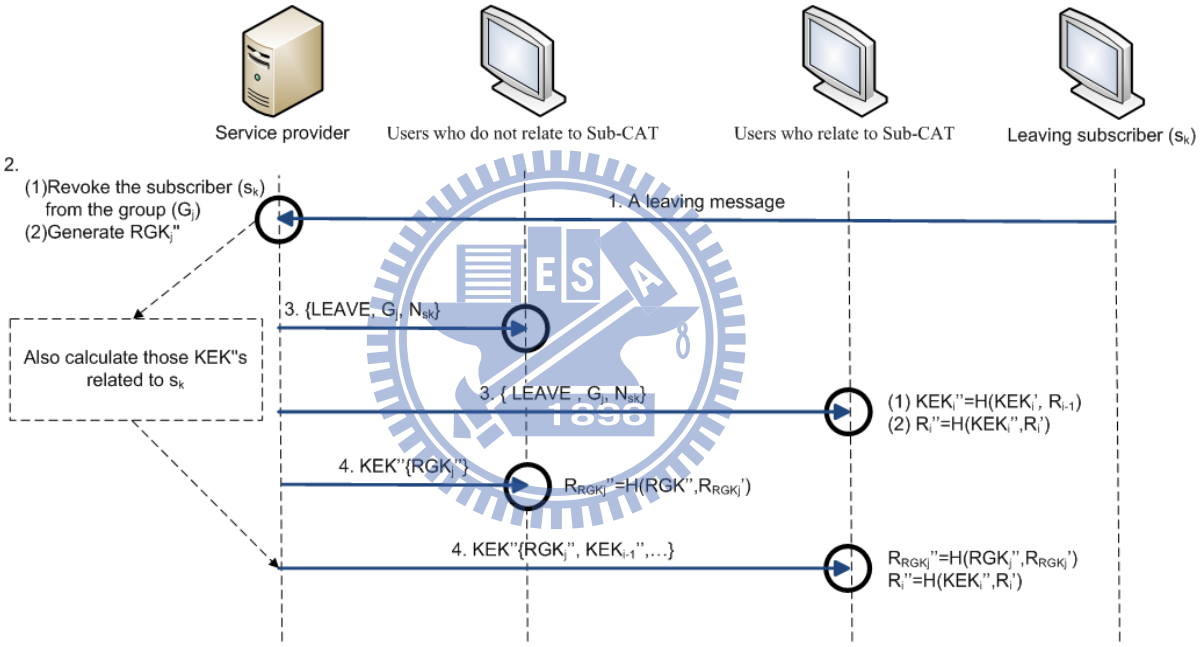


Figure 21: Leave Operation

Figure 22 is a clearer example. When a user (u3) stops to subscribe any channel, Leave Operation is applied to refresh the group key (RGK_1''), key encryption keys (KEKs), and R values. Service provider broadcasts a leaving message to group (G_1), {LEAVE, G_1, N_{u3}}. Then, group members relating to Sub-CAT automatically update keys. u1, u2 automatically update KEK_{14}' and R_{14}' value, KEK_{14}'=H (KEK_{14}, R_{12}), R_{14}'=H (KEK_{14}', R_{14}). u4 updates KEK_{34}' and R_{34}' value, KEK_{34}'=H (KEK_{34}, R_4), R_{34}'=H (KEK_{34}', R_{34}). After those steps, the keys' relations are broken. Service

provider needs to multicast new RGK and KEK to users who held those before updating. Service provider transmits messages: KEK_{58} {RGK_1'}, KEK_{14}' {RGK_1'}, KEK_{34}' {KEK_{14}', RGK_1'}. The R values are also updated after group members receive messages. u4 needs to calculate R_{14}'=H (KEK_{14}', R_{14}) and u5, u6, u7, and u8 need to calculate R_{RGK1}'=H (RGK', R_{RGK1}). Finally, the CAT is updated.

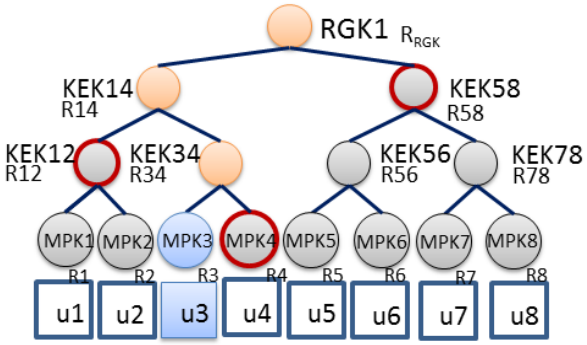


Figure 22: Key structure of Leave operation (1)

Multi-leave is an efficient choices to service providers, and it is also called batch operation. When more than one subscriber leaves the group, the leave operation is a little different with only one subscriber leaving a group. If one interior node's both children leaf nodes are updated, server generates new key and transmits to those leaf nodes. As shown in Figure 23, when subscribers (u1 and u3) simultaneously leave group (G_1):

1. Service provider broadcasts {LEAVE, N_{u1}, N_{u3}, G_1}.
2. Group members update KEKs and R values as describing before.
   - u2 updates KEK_{12} and R_{12}, KEK_{12}'=H ($KEK_{12,}$ R_2), R_{12}'=H (KEK_{12}', R_{12}).
   - u4 updates KEK_{34} and R_{34}, KEK_{34}'=H (KEK_{34}, R_4), R_{34}'=H (KEK_{34}', R_{34}).
3. Service provider needs to generate not only RGK' but also KEK_{14}'. KEK_{14}' needs to generate by service provider, because node12 and node34 both

updated.

After those steps, the rest steps are same with the original situation that only one subscriber leaving the group. Because of the keys' broken relations, service provider needs to multicast new RGK and KEK to users who held those before updating. Service provider sends a KEK58 {RGK_1'} to u5, u6, u7 and u8, KEK_{12}' {RGK_1', KEK_{14}'} to u2, and KEK_{34}' {RGK_1', KEK_{14}'} to u4. Then, the R values corresponding to RGK and KEKs are updated by themselves, R_{RGK1}'=H (RGK_1', R_{RGK1}) and R_{14}'=H (KEK_{14}', R_{14}).

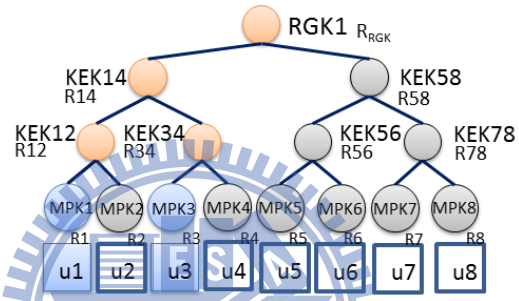


Figure 23: Key structure of Leave operation (2)

In this way, IPTV service provider could prevent the subscriber from watching channels, and provides the backward secrecy.

3.4.3 Change Operation

Change Operation is used when an old subscriber changes the group. As show in Figure 18, when an old subscriber, S9, decides to subscribe different channels, he/she leaves G1 and joins G2. The join operation and leave operation are used for updating keys: RGK_1, RGK_2, $KEKs_{Group1}$, $KEKs_{Group2}$, AKs, $KEKs_{Channel}$, Rs_{Group1}, Rs_{Group2}, $Rs_{Channel}$.

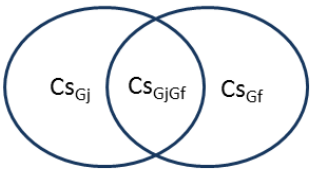


Figure 24: A channel group

There channel trees are classified to three parts, when members change their subscriptions and groups. Fist part is those channels which members do not subscribe any more. Second part is those channels which members still subscribe. Third part is those new channels which members are going to subscribe. Take Figure 24 as an example. The symbol "Cs_{Gj}" means a set of channels G_j subscribes. The symbol "Cs_{Gf}" means a set of channels G_f subscribes. The symbol "Cs_{GjGf}" means a set of channels both G_j and G_f subscribe.

When a user (u_k) changes his/her group from G_j to G_f, those channels' key's must are updated in "A" and "B". Because of both forward and backward secrecy, those channels in "A" must do "Leave operation", and those channels in "B" must do "Join operation". However, the channel trees' key encryption keys in "C" are possibly suffered from collusion attack. Therefore, AKs, $KEKs_{Channel}$ and $Rs_{Channel}$ are must updated. Lots of group trees are included in channel trees, Figure 9.

When u_k changes his/her group from G_j to G_f, the processes of change operation in channel tree and group tree are following. Those processes in channel tree are clearly shown in Figure 24 and Figure 25: However, the processes of change operation in group tree are specifically described in the leave operation and join operation.

1. Server broadcasts {CHANGE, N_{uk}, G_j, G_f}
2. Rest members in G_j do the operation leave. Group members in G_f do the join operation.
3. Updating keys in channel trees.
    1). Those groups in channel trees of Cs_{Gj} do "Leave Operation". The keys in channel trees will be updated by groups who subscribe same channels with G_j

2). Those groups in channel trees of Cs_{Gf} do “Join Operation”. The keys in channel trees will be updated by groups who subscribe same channels with G_f.

3). Those groups in channel trees of Cs_{GjGf}

- If groups’ corresponding nodes relate to $CAT_{Gj\&Gf}$ of channel tree, those nodes’ corresponding keys do not need to be updated. The $CAT_{Gj\&Gf}$ is those nodes which both in CAT_{Gj} and CAT_{Gf}, when CAT_{Gj} and CAT_{Gf} are on a same channel tree.
- If groups’ corresponding nodes do not relate to $CAT_{Gj\&Gf}$, nodes’ corresponding KEKs and Rs need to be updated.

  i. If groups’ corresponding nodes relate to CAT_{Gj}-$CAT_{Gj\&Gf}$, groups do “Leave Operation”. CAT_{Gj} is those corresponding nodes on the channel tree of ancestors and affected leaves, when G *j* does the join operation or leave operation.

  ii. If groups’ corresponding nodes relate to CAT_{Gf}-$CAT_{Gj\&Gf}$, groups do “Join Operation”. CAT_{Gf} is those corresponding nodes on the channel tree of ancestors and affected leaves, when G *f* does the join operation or leave operation.

4. Server unicasts new keys to u_k. MPK_{uk} {RGK_f’,$KEKs_{Group}$’, R_{RGKf}’, Rs_{Group}’, AKs’, $KEKs_{Channel}$’ , Rs_{AK}’, $Rs_{Channel}$’}

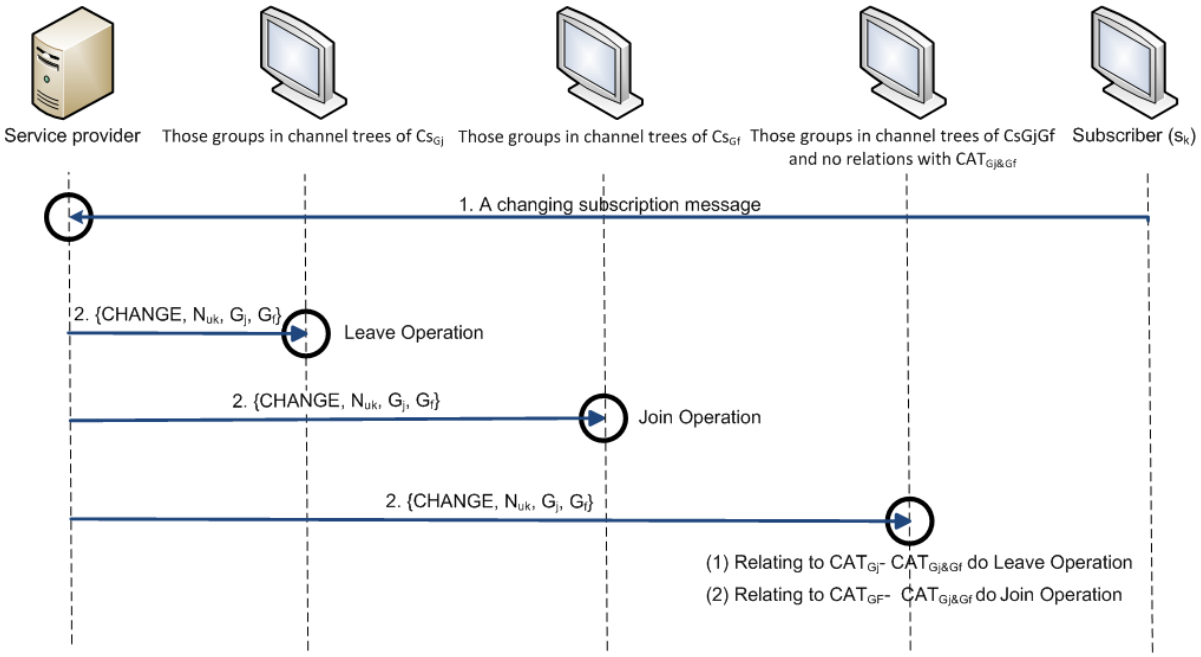


Figure 25: The change operation in channel tree

More specifically in third process, assume that channel number 1 (AK1) belongs to "Cs_{GjGf}", user k (uk) changes group from group number 1 (G_1) to group number 3(G_3) in Figure 26. Firstly, in channel tree number 1, KEK_{12} and R_{12} are updated by the leave operation, due to the nodes is on the path of G_1 and not on the path of G_3 to the root, CAT_{Gj}- $CAT_{Gj\&Gf}$. Then, G_2 are going to calculate KEK_{12}'= H (KEK_{12}, R_{RGK2}) and R_{12}'= H (KEK_{12}', R_{12}). Service provider multicasts KEK_{12}' and R_{12}' and encrypts those by RGK_1', RGK_1' { KEK_{12}', R_{12}'}. KEK_{34}, however, is on the path of G_3 to the root. The node corresponding to KEK_{34} and R_{34} are going to do the join operation, CAT_{Gf}- $CAT_{Gj\&Gf}$. KEK_{34} and R_{34} is updated through KEK_{34}'=H (KEK_{34}, R_{34}) and R_{34}'=H (KEK_{34}', R_{34}). The other nodes' corresponding keys do not need to change. The third process is also shown in Figure 25.

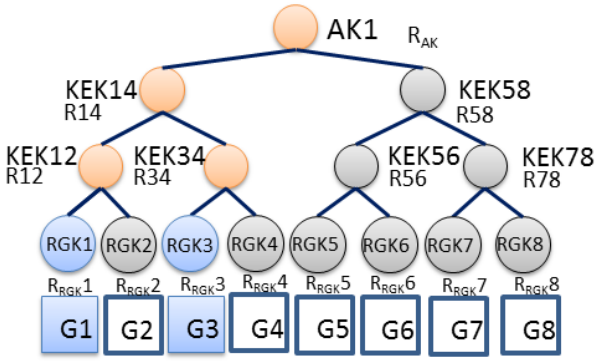


Figure 26: Key structure of change operation

3.4.4 Per_update Operation

Moreover, if the group membership in a group is not changed in a period of time, Per_update Operation is used to update the subscribers' group key and channel keys. The group key is encrypted by the subscribers' old group key or channel keys. After that, each subscriber updates their own administration keys and R values. For example, when channel keys are updated, AK {AK'} is used. It's the same that group keys are updated through RGK {RGK'}. By using Per_update, the risk of cracking keys is going down.

3.5 Balance Tree: A Problem of the Proposed Protocol and its solutions

The possible problem is that a tree easily becomes unbalanced, when lots of group members join and leave the group. Key management is inefficient when a tree is unbalance. Service manager also transmits more multicasting messages when the tree is unbalance. Therefore, keep trees in balance is needed. In this section, there are two operations to solve a possible problem of this proposed protocol: Multi-LeaveNode Operation and Multi-JoinNode Operation.

For efficiently key management in tree structure, binary searching tree is implemented in this thesis. There is more efficient in searching, adding, and deleting certain node, when implement a binary searching tree. Besides, there are two operations to manage keys and

maintain a tree's balance: Multi-LeaveNode Operation and Multi-JoinNode Operation. Multi-LeaveNode Operation is used when more than half of tree's leaf nodes are vacant. Multi-JoinNode Operation is used when the number of group subscribers is more than the number of tree's leaf nodes. Following sections are going to describe in detail.

3.5.1 Multi-LeaveNode Operation

When numbers of vacant leaf nodes are half more than all tree's leaf node, Multi-LeaveNode Operation is trigger by service providers for efficient key management. Multi-LeaveNode Operation also reconstructs a smaller tree and minimizes the service manager's loading as much as possible. Service manager's computation and transmitting messages are increasing, when service manager reconstruct a tree and new keys. Hence, minimize the service manager's loading also is a key issue. For those issues have described, there are two parts in this section: first is maintaining a tree's balance and second is key updated when a tree's structure is changed.

In Multi-LeaveNode Operation, there are processes to determine the method deleting nodes. A new tree then needs to be constructed and add leaf nodes depending on the tree's key structure. In other word, those processes are:

1. A Service manager checks that is there a sibling node of the tree's each vacant leaf nodes, when choosing the deleted nodes.
2. Service manager updates keys.
3. Service manager checks each interior node.
4. Reconstruct a new tree and move the old tree to a new tree. Add new leaf nodes if necessary.
5. Update keys again.

In the first and second procedures, if there is a sibling node of the vacant leaf node, service provider deletes the vacant leaf node and its upper level interior node. The

steps are described as following, Figure 27:

1. Delete $N_{Leaving}$, $N_{Leaving\ -1}$
2. Do “Leave Operation”:
   - Update $KEK_{Leaving-2}$ and $R_{Leaving-2}$
3. Service provider sends $MPK_{s\_Leaving}$ $\{KEK_{Leaving-2}, R_{Leaving-2}\}$

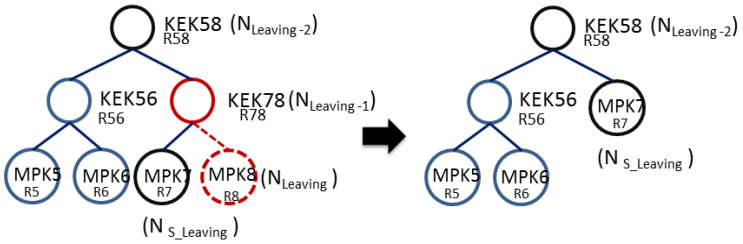


Figure 27: Multi-LeaveNode Operation (1)

There is an example clearly explained in Figure 27: When delete the leaving node ($N_{Leaving}$) corresponding to MPK 8, delete upper level node of leaving node ($N_{Leaving-1}$) corresponding to KEK78, too. Then Leave Operation is triggered, then the two upper level node of leaving node ($N_{Leaving-2}$) corresponding to KEK58 is updated through KEK58’=H (KEK58, R56), and R58 is also updated along with KEK58, R58’=H (KEK58’, R58). However, $N_{Leaving-1}$ don’t need to be updated, and it already was deleted before. Service provider needs to send new messages to the sibling node of leaving node ($N_{s\_Leaving}$) corresponding to MPK7. The message includes the KEK and R value of two upper level node of leaving node (KEK58’ and R58’).

If the sibling node of vacant leaf node is also vacant, service provider deletes four nodes: the vacant leaf node, its sibling node and its upper levels’ two interior nodes. The steps are described as following, Figure 28:

1. Delete $N_{Leaving}$, $N_{s\_Leaving}$, $N_{Leaving\ -1}$, ,and $N_{Leaving-2}$
2. Do “Leave Operation”:
   - Update upper levels KEK and R value if there is necessary.
3. Service provider sends $KEK_{s\_Leaving-1}$ $\{KEK_{upper\_levels}, R_{upper\_levels}\}$

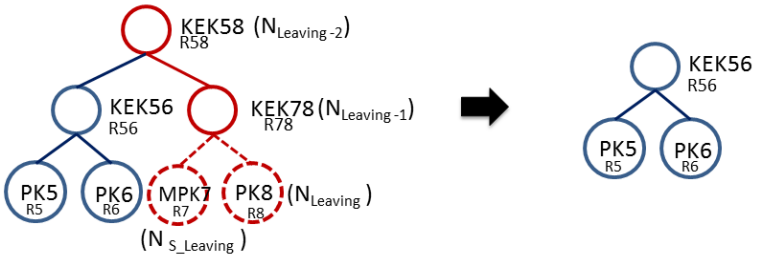


Figure 28: Multi-LeaveNode Operation (2)

As shown in Figure 27. When delete the leaving node ($N_{Leaving}$) corresponding to MPK 8, its sibling node ($N_{s\_Leaving}$) corresponding to MPK7, and upper levels' two nodes of leaving node ($N_{Leaving-1}$, $N_{Leaving-2}$) corresponding to KEK78 and KEK58, too. Then Leave Operation is triggered, if it is a subtree in Figure 28. Then the processes of Leave Operation are same with that there is a sibling node of vacant leaf node.

After first and second steps, service manager will check each interior node. If there is/isn't branches the way to update key and delete node are same with above. If there are two branches, there are no actions. If there is only a branch, the interior node needs to be deleted. Its upper node's key needs to be updated by the R value of the interior node's branch node. If there is no branches, service manager needs to delete the interior node, and check upper two level nodes. This process will keep in loop until the node is root.

In the fourth procedure, the new tree needs to be reconstructed. After deleting nodes, the tree easily become unbalance, and construct a new tree is needed. A full balanced tree is supposed to build, and the size of the tree is according to the rest member in group. For example, the numbers of group member are five, then the height of full balanced tree is 4, ceil ($\log_2 5$)+1. The tree is binary searching tree and each node's number is arranged in order as described before. Service manager moves old tree to the new tree in order. However, there is a circumstance that two leaf nodes are arranged together. Service manager adds a new interior node between those two leaf nodes, and keep the tree from disorder.

In the last process, keys are updated more frequently if the original tree‘s order is in a mess. After a balance tree constructed, the key structure will check by service manager. The ways to update keys are measuring each interior nodes’ parents are the same with in original tree. If each interior node from leaf node to root is not same after rebalance, those interior nodes update by one of its branch’s R values. Then service provider transmits those new keys to users who should hold those keys.

3.5.2 Multi-JoinNode Operation

When numbers of joiners are more than the tree’s vacant leaf node, Multi-JoinNode Operation is used to construct a bigger tree. Another goal of Multi-JoinNode Operation is maintaining keys still usable in the group, and cutting down the service loading as much as possible. There are also two parts in this section: first is maintaining a tree’s balance and second is key updated when a tree’s key structure is extended.

In Multi-JoinNode Operation, a full balanced binary searching tree is supposed to build. The way to build a new tree is similar with the new tree reconstruction in Multi-LeaceNode Operation. The size of the new tree depends on the all numbers of group members. After a new tree built, the original tree supposes to be the subtree of the new tree. The processes are:

1. Make a new tree whose height is ceil ($\log_2 N_{All\_members}$). The $N_{All\_members}$ is the total numbers of group member, and those numbers are summation from the original group members and new joiner, $N_{All\_members} = N_{original\_members} + N_{joiner}$.
2. Move old tree to a new tree in order. Old tree becomes the left subtree of new tree.
3. The keys are managed and changed:
   - Service provider transmits $KEK_{OldRoot}$ {$KEK_{NewRoot}$, $R_{NewRoot}$}, and updates keys in the original key structure.

- Service provider unicasts keys and relations to each joined nodes.

Take Figure 29 as an example. Assume that the height of original tree is three and the tree contains at most four members. Multi-JoinNode Operation is triggered, when the original tree is full of members and there are still more than two members need to join in the group. The original tree is shown as the left part in Figure 29 in red color. According to those processes have described above, $N_{All\_members}$ are six. And the height of new tree is four, ceil ($\log_2 6$). In step two, the old tree structure is moved to new tree as a left subtree in Figure 29. The original keys in red don't be updated. Each old tree's members have another new keys and service provider transmits KEK14 {KEK1-8, R1-8} to them. The rest two members are placed in the right subtree's vacant leaf nodes in order. The KEKs and R values in black color are generated by service provider. Service provider unicasts each new tree member keys and R values encrypted by his/her MPK.

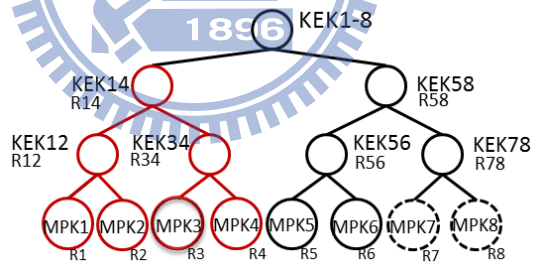


Figure 29: Multi-JoinNode Operation

In this way, service provider doesn't need to generate all new keys for a new tree and unicast all members. Service provider only needs to transmit three packages and generate three keys (KEK1-8, KEK58, and KEK56) and corresponding relations.

3.6 Discussions

This section analyzes those issues mentioned in section 3.1 by using channel-based key management, and shows that those problems do not exist.

- **Refreshments of keys and R values:**

In Sun's et al. CAS, subscribers are not allowed to rejoin a group. In another circumstances, subscribers must be in the same node or path when rejoin the same tree, because those information he/she held before are not updated yet.

Preventing from forward and backward secrecy problems in this thesis, there are rekeying operations whenever users join/ change/ leave a group. Updating keys and R values also contribute the solid key structure and avoid collusion attacks. Therefore, subscribers could rejoin a tree and be at any node in a tree.

Here is an example. When a user (u3) leaves a group (G1), those processes are following, Figure 22. After those steps, u3 still couldn't break the key structure.

1. service provider broadcasts {LEAVE, N_{u3}, G1}
2. *u1,u2,u4* automatic update KEKs, R values:
   - *u1,u2* update *KEK14* and *R14,* through *KEK14'=H(KEK14,R12)* and *R14'=H(KEK14',R14)*
   - *u4* updates KEK34 and R34, KEK34'=H (KEK34,*R4)*, *R34'=H(*KEK34' ,*R34).*
3. Service provider multicasts RGK' to users through messages, *KEK58 {RGK1'}, KEK14'{RGK1'}, KEK34'{KEK14', RGK1'}.*
4. *u4* automatically updates R14, *R14'=H(KEK14',R14)*. All rest group members also automatically update R_{RGK}'=H (*RGK1',* R_{RGK}).

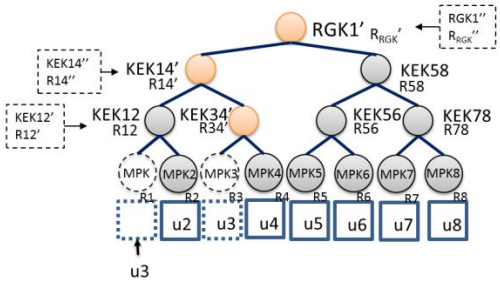


Figure 30: Key structure of user rejoining

After keys and R relations are updated, the key structure shows in Figure 30.

When the user (u3) rejoins the group in different leaf node (first place), all keys and relations are automatically updated by members:

1. Service provider broadcasts {Join, N1, G1}, and group members update RGK1 through RGK1''=H (RGK1', R_{RGK}')
2. Each user relates to node12, node14 automatically update KEK12 , KEK14, R12, and R14:
   - KEK12'=H(KEK12,R12), KEK14''=H(KEK14',R14')
   - R12'=H(RGK',R12),R14''=H(RGK',R14')
3. Service provider unicasts to u3 a message, PK1 {KEK12', KEK14'', RGK1'', R1', R12', R14'', R_{RGK}''}.

- **Collusion attack prevention:**

There are two situations when collusion attack happened. First one is that more than two users simultaneously leave from same group. Each user only has the keys and R relations on the path from root to the node corresponding to him/ her. The original key structures and relations are broken after Leave Operation triggered. Therefore collusion attack will not happen. Figure 23 is an example and shows in multi-leave operation.

Another situation is that users leave a group one after the other. Because there are join, leave, and change operation, the keys are hard to broken in this situation. In Figure 23 as an example, when u1 follows up u3 leaving the group G1, the processes are following:

- When u3 leave, service provider broadcasts {LEAVE, N_{u3}, G1} in Figure 22.
  1. *u1,u2,u4* automatic update KEKs, R values:
     - *u1,u2* update *KEK14* and *R14,* through *KEK14'=H(KEK14,R12)* and *R14'=H(KEK14',R14)*
     - *u4* updates KEK34 and R34, KEK34'=H(KEK34,*R4),*

*R34'=H(*KEK34' *,R34).*

2. Service provider multicasts RGK' to users through messages, *KEK58 {RGK1'}, KEK14'{RGK1'}, KEK34'{KEK14', RGK1'}.*
3. *u4* automatically updates R14, *R14'=H(KEK14',R14)*. All rest group members also automatically update R_{RGK}'=H (*RGK1',* R_{RGK}).

- When u1 follows up u3 leaving the group G1, service provider broadcasts {LEAVE, N_{u1},G1} in Figure 31
  1. *u2,u4* automatically update KEKs, R values:
     - *u4* update *KEK14* and *R14*, through *KEK14"=H(KEK14',R34')* and *R14"=H(KEK14",R14')*
     - *u2* updates KEK12 and R12, KEK12'=H (KEK12, *R2), R12'=H (*KEK12', *R12).*
  2. Service provider multicasts RGK' to users through messages, *KEK58 {RGK1"}, KEK14"{RGK1"}, KEK12'{KEK14", RGK1"}.*
  3. *u2* automatically updates R14, *R14'=H(KEK14',R14)*. All rest group members also automatically update R_{RGK}"=H (*RGK1",* R_{RGK}').

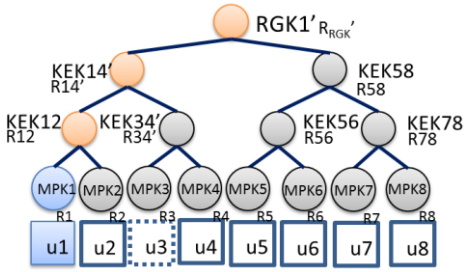


Figure 31: Key structure of Leave operation (3)

After those steps, the key structure is showing in Figure 32. Even though, u3 has RGK1, KEK14, KEK34, MPK3, R4, R_{RGK}, R14, and R34. u1 has RGK1', KEK14', KEK12, MPK1, R1, R_{RGK}', R14', and R12. There is no way to trace the new keys or R values, if u1 and u3 work together. However, in Sun's et al. CAS, there are risks of collusion attacks in both situations described above.

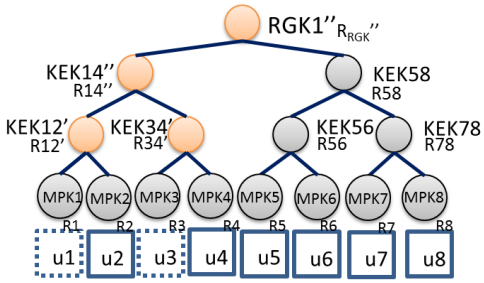


Figure 32: Key structure of Leave operation (4)

- **Tree rebalances:**

The rebalance situations do not mentioned a lot in Sun's et al. CAS, and only extension the tree's size is included. There are balance operations in this thesis. Those are triggered, when the tree is not balanced or group members are more than the tree's capacity. Service provider transmits messages and manages group members more efficiently after implementing those rebalancing operation.

Chapter 4: Simulation and Security Analyses

This thesis mainly proposes a key management suitable for IPTV services, which is also called IPTVP. IPTVP solves those problems in Sun's et al. CAS in chapter three. This chapter is going to compare and analyze security and simulation results and analyses between Sun's et al. CAS, group key protocols and the protocol this thesis proposed. There are abbreviations and acronyms using in this chapter and showing in Table 6.

Table 6: Abbreviations and acronyms (2)

Abbreviations and acronyms			
SKDC	Simple key distribution center	**n**	Number of group members
LKH	Logical key hierarchy	**M_L**	Number of leaving members
OFT	One-way function tree	**M_J**	Number of joining members
IPTVP	IPTV protocol (this thesis's protocol)	**C_E**	Encryption costs
S_Ccomp	Service manager's computational costs	**C_f**	Hash function costs
M_Ccomp	Each member's computational costs	**C_K**	Hash function costs to generate a key
Msg	Number of messages	**C_R**	Hash function costs to generate a relation
S_St.	Service manager's storages	**sl**	The size of CAT
M_St.	Each member's storages		

***$C_f = C_r = C_K = C_R$**

4.1 Security Analysis

This section is going to analyze security of the method proposed in this thesis. Those security issues are mentioned in section 2.2. This thesis focuses on the dynamic group in IPTV environment. Subscribers could join/leave/change a group whenever they like. Those security requirements are:

- Confidentiality: Those data/videos/voices are transmitted in cypher text in IPTV environment. The keys are used to encrypt those packages, and only members held those keys can get decrypt text. Hence, illegal users, who do not belong to the group, could not get the messages.
- Authentication: There are subscriber register phase and channel subscribing phase to authorize users'. Those phases could prevent unauthorized user receiving and getting contents.
- Backward and forward secrecy: Rekeying is important for service managers to provide both securities. There are join, leave, and change operations rekeying members' keys and collusion attacks prevention in this thesis. More detail contents are described in previous chapter.

According to those security requirements, there are comparisons with Sun's et al. CAS and the protocols this thesis proposed, Table 7. Obviously, this thesis enhances the securities, including providing both forward and backward secrecy and collusion attacks prevention. Besides, as shown in Table 8, IPTVP is also as strong as other group protocols.

Table 7: Security analyses with Sun's et al. CAS

	Confidentiality	Authentication	Forward secrecy	Backward secrecy	Collusion Attack
IPTVP	Y	Y	Y	Y	Y
Improve Sun's et	Y	Y	Y	Y	N

al. CAS					
Sun's et al. CAS	Y	Y	N	N	N

Table 8: Security analyses with group key protocols

	Forward secrecy	Backward secrecy	Collusion Attack
SKDC	Y	Y	Y
LKH	Y	Y	Y
OFT	Y	Y	Y
IPTVP	Y	Y	Y

4.2 Simulation Results and Analytical Analysis

There are two parts in this section. First part is going to analyze the costs from simulation results in a quantitative way when rekeying operations triggered. Second part is the simulation results showing that maintaining balance trees in two situations as examples. The simulation programs are writing in C++ and simulation environment is Windows operating system.

4.2.1 Simulation Results and Analytical Analysis

Cutting down delay time is critical issue when users start using services. Besides, with rapid development of internet, users are easier work together and break keys. To avoid collusion attacks, keys are need to frequently update and manager's loading are increase, too. Therefore, there are three indexes and five scenarios to analyze the efficiency of protocols in analytical analyses.

This thesis mainly modified from Sun's et al. CAS. Sun's et al. CAS 's feature is that users have flexibility to subscribe/ unsubscribe the service. Dynamic groups cause

key structure changing more frequently, and keys are harder to manage. This thesis's protocol is stronger, due to preventing more security issues described in section 4.1. On the other hand, service providers' loading is also an issue. There are some indexes to evaluate the efficiency when a member joining or leaving a group:

- **Computational Costs:**

  When a member joins/leaves a certain group, the keys are going to update and generate by service provider. The service manager's computational costs include generate new keys and encrypting messages. Those two items are factors that affect the efficient key management.

- **Number of rekeying messages**

  Rekeying messages are needed, when a member joins/leaves a group. When there are less rekeying messages, there are less bandwidth costs and time to encrypt/decrypt messages.

- **Storages**

  There are keys managing by service provider in this protocol. Those keys are including group key, acknowledge keys, key encryption keys, R values…etc. As long as those keys decreasing, the storage costs are less to manager or each user.

Multicasting is a way to manage keys and transmit keys encrypted by KEK. It consumes lots of time to updates keys, when members frequently join or leave a group. Numbers of rekeying messages is an index to measure the efficiency of updating keys and encrypting messages. If the group size is large and has lots of keys, rekeying messages' time costs and computational costs are tremendous and could not be ignored.

This section is focus on measuring rekeying messages, storages, and computational costs, and comparing with other protocols, like Simple Key Distribution Center (SKDC), Logical Key Hierarchy (LKH), and One-way Function Tree (OFT)[12][15]. The ways to

update keys are different when member joins and leaves a group. Therefore, there are four scenarios analyzing rekeying messages and computational costs: the join operation triggered, the leave operation triggered, the multi-join operation triggered, and the multi-leave operation triggered. Then, there is a scenario compares storages between IPTVP, SKDC, and LKH. The costs are summarized C_E, C_f, C_K, and C_R. Following clearly describe factors' analytical analyses in those five scenarios.

4.2.1.1 Scenario 1: The join operation triggered

There are two factors to analyze the efficiency between protocols in this scenario: number of rekeying messages and computational costs.

When a member joins a group, the costs to rekey are shown in Table 9. IPTVP only needs to transmit a plain text to inform related member updating keys when a member joins a certain group. Other group protocols transmit new keys to group members encrypted by old keys when a member joins the group. On the other hand, group members need to compute in IPTVP than those group protocols.

Table 9: Efficiency comparisons when a member joins

	SKDC	**LKH**	**OFT**	**IPTVP**
Msg	n	$2(\log_2 n)$	$\log_2 n$	1
S_Ccomp	C_E*n	$C_E(2\log_2 n)$	$C_E(\log_2 n)+2C_f(\log_2 n)$	1
Max. M_comp	C_E	$C_E(\log_2 n)$	$C_E+C_f(\log_2 n)$	$(\log_2 n)(C_K+C_R)$
Min. M_comp	C_E	C_E	C_E+C_f	C_K+C_R

The quantitative comparisons show in Figure 33 and Figure 34. In Figure 33, IPTVP broadcasts only a message to all members no matter how many group members are there. SKDC, however, needs to send n of messages, when there are n of group members. OFT and LKH needs to send $\log_2 n$ or $2\log_2 n$ messages, but those messages still are increasing according to group members.

In Figure 34, the managers' computational cost is only one in IPTVP, and the reason is same with number of messages transmitted when joining a member. OFT, LKH, SKDC are increasing computation costs as group member growing.

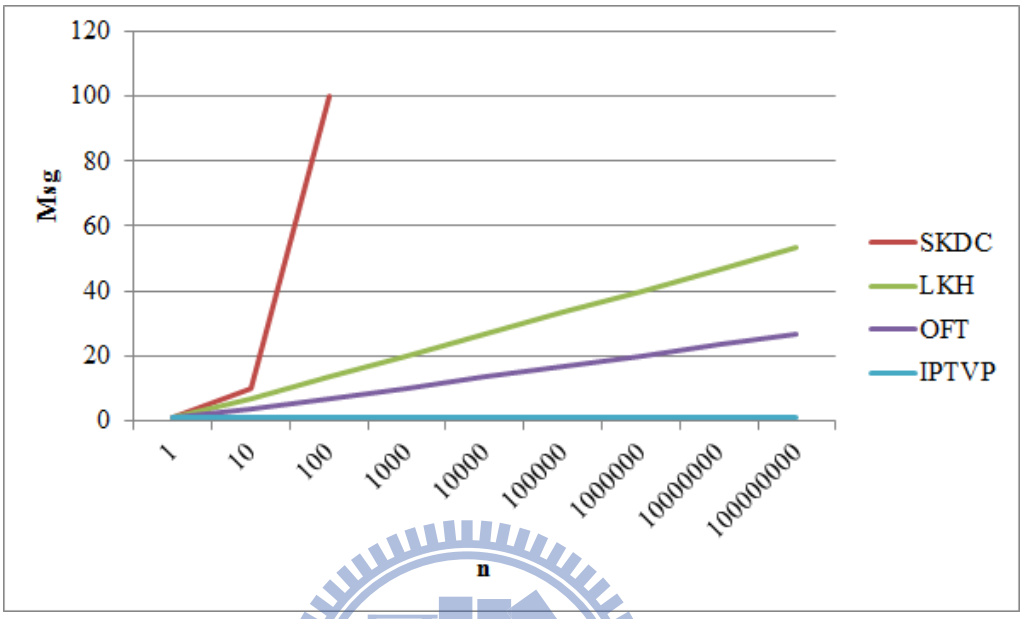


Figure 33: Number of rekeying messages when a member joins a group

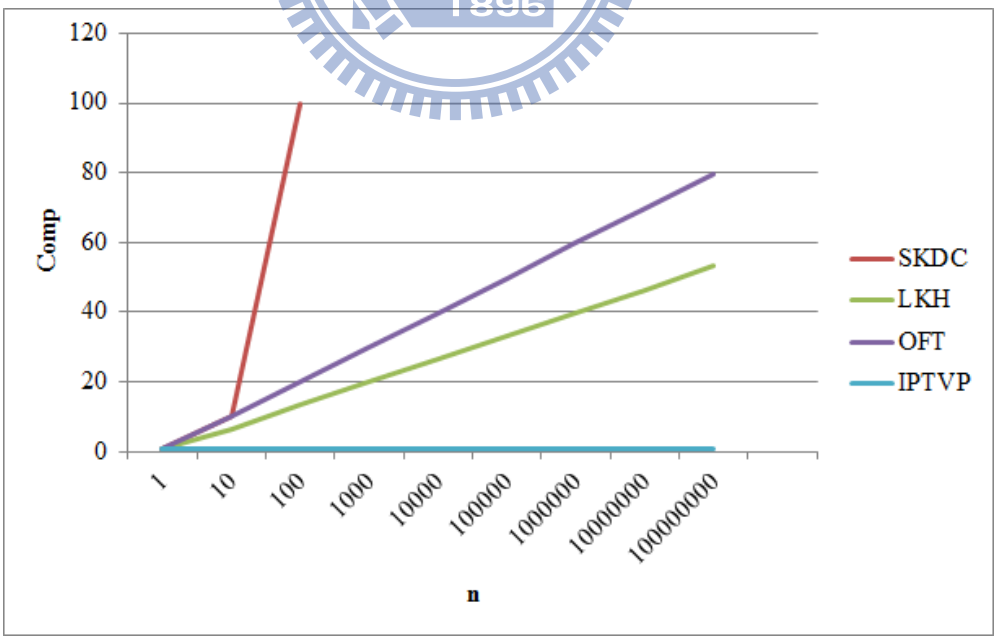


Figure 34: The computational costs of service manager when a member joins a group

4.2.1.2 Scenario 2: The leave operation triggered

There are two factors to analyze the efficiency between protocols in this scenario: number of rekeying messages and computational costs.

When a member leaves a group, the costs to rekey are shown in Table 13. IPTVP transmits new keys to those members held the keys before and encrypts keys by KEK. The way is same with other group protocols, SKDC, LKH, and OFT. IPTVP transmits $\log_2 n$ numbers of messages, and it is same performance with OFT. LKH needs to send twice numbers of messages than OFT and IPTVP. SKDC is a worst situation. In Figure 35: Number of rekeying messages, there is a quantitative figure showing the number of messages transmitted when a member leaves.

In Figure 36, IPTVP manager' computational costs are the least. IPTVP only needs to encrypt the messages when a member leaves the group. OFT also encrypt the messages, but OFT needs to generate blind keys. LKH encrypts the messages, but those messages need to transmit are twice than IPTVP, $2(\log_2 n)$, when a member leaves a group. SKDC also is a worst situation

Table 10: Efficiency comparisons when a member leaves

	SKDC	**LKH**	**OFT**	**IPTVP**
Msg	N	$2(\log_2 n)$	$\log_2 n$	$\log_2 n$
S_Ccomp	nC_E	$C_E(2\log_2 n)$	$C_E(\log_2 n)+2C_f(\log_2 n)$	$C_E(\log_2 n) +C_K(\log_2 n -1)$
Max. M_comp	C_E	$C_E(\log_2 n)$	$C_E+C_f(\log_2 n)$	$C_K+C_R+C_{E+}(\log_2(n-1))(C_R)$
Min. M_comp	C_E	C_E	C_E+C_f	C_R+C_E

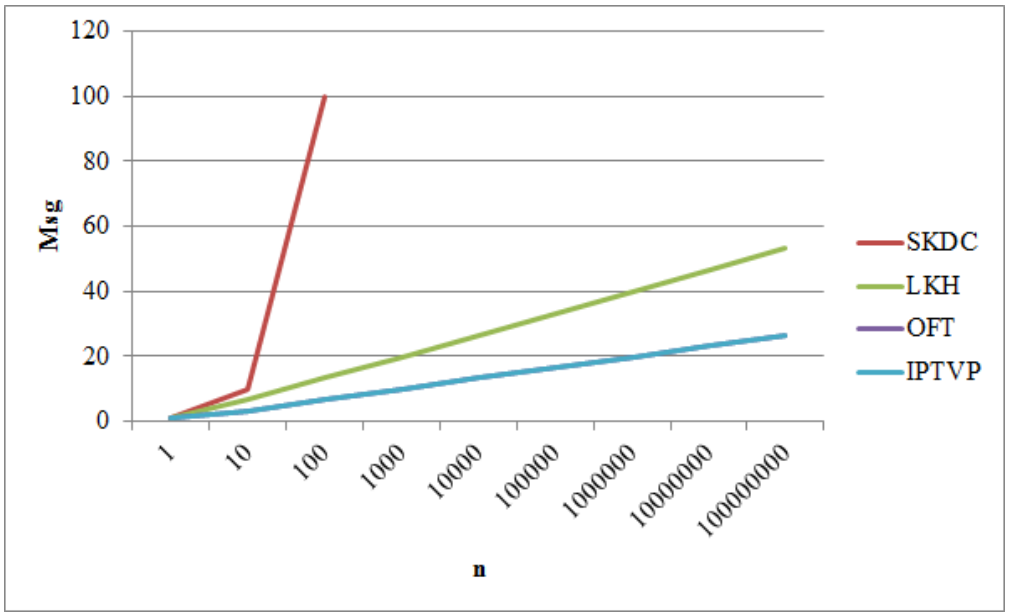


Figure 35: Number of rekeying messages when a member leaves a group

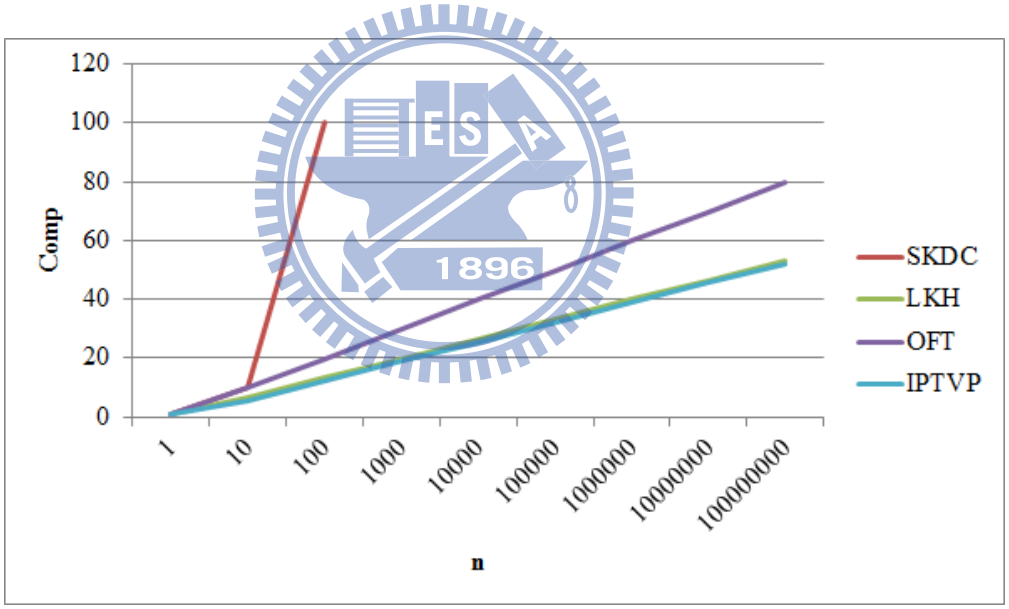


Figure 36: The computational costs of service manager when a member leaves a group

4.2.1.3 Scenario 3: The multi-join operation triggered

There are two factors to analyze the efficiency between protocols in this scenario: number of rekeying messages and computational costs.

When more than one member join a group in same time, the costs to rekey are shown in Table 11. IPTVP only needs to transmit a plain text to inform related member

updating keys when members join a certain group. However, other group protocols transmit new keys to group members encrypted by old keys when a member joins the group. On the other hand, group members need to compute in IPTVP more than those group protocols.

Table 11: Efficiency analyses when members join a group

	SKDC	**LKH**	**OFT**	**IPTVP**
Msg	n+l	2*sl*	*sl*	1
S_Ccomp	$(n+l)C_E$	$C_E(2sl\text{-}l)$	$C_E*sl+2C_f*(sl\text{-}l)$	1
Max. M_comp	C_E	$C_E(\log_2 n)$	$C_E+C_f(\log_2 n)$	$(\log_2 n)(C_K+C_R)$
Min. M_comp	C_E	C_E	C_E+C_f	C_K+C_R

The quantitative comparisons show in Figure 37 and Figure 38. Both in Figure 37 and Figure 38 assume that there are 1024 group members, and random members join a group in a same time. The results are average number from 2000 set random members

In Figure 37, IPTVP broadcasts only a message to all members no matter how many group members join the group in the same time. SKDC, however, needs to send n+l of messages, when l members join a group in a same time. OFT and LKH needs to send 2*sl or sl messages, and are more than IPTVP.

In Figure 38, the managers' computational cost is only one in IPTVP, and the reason is same with number of messages transmitted when members join a group in a same time. OFT, LKH, SKDC are increasing computation costs as group member growing or numbers of joiners increasing.

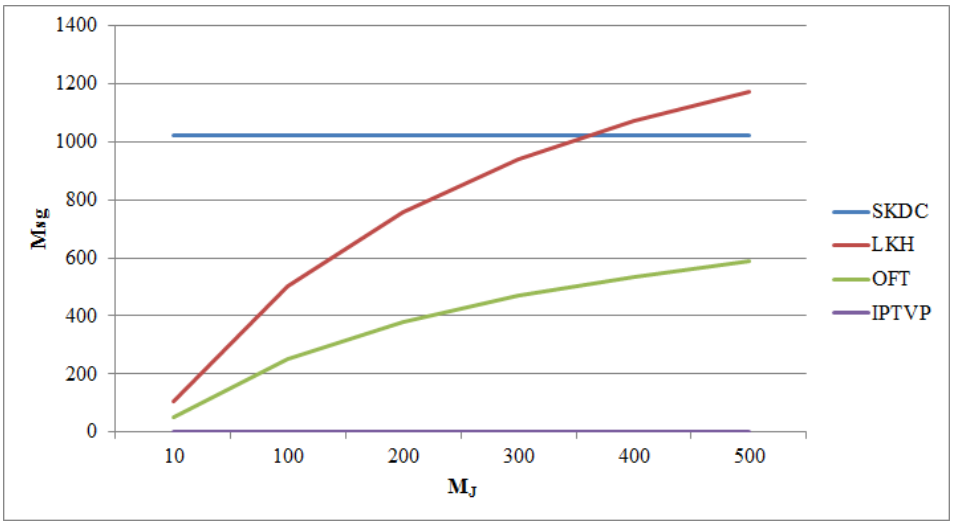


Figure 37: Number of rekeying messages when members join a group

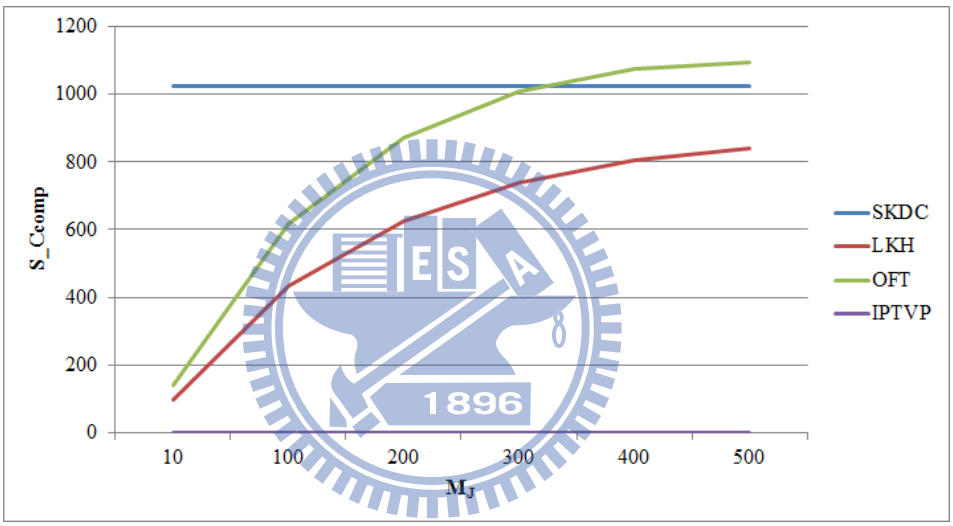


Figure 38: The computation costs of service manager when members join a group

In Figure 37 and Figure 38, there are curve lines instead of straight lines. The reason is some interior nodes' corresponding administration keys do not sent by managers. If an interior node' all leaves' members are new joiners, there is not necessary to send rekey messages to group members. Managers, instead, send those news keys by unicast to each new user. However, SKDC don't use administration keys. Hence, SKDC shows a straight line in simulation result figures.

4.2.1.4 Scenario 4: The multi-leave operation triggered

There are two factors to analyze the efficiency between protocols in this scenario: number of rekeying messages and computational costs.

When members leave a group in a same time, the costs to rekey are shown in Table 13. IPTVP transmits new keys to those members who held the keys before and encrypts keys by KEK. The way is same with other group protocols, SKDC, LKH, and OFT. IPTVP transmits *sl*-*l* numbers of messages, if there are *l* members leave the group in a same time. However, OFT and LKH needs to send 2*sl*-l and 2*sl* numbers of messages, and are more than IPTVP. SKDC is not a worst situation in this case, and it needs to send n-l numbers of messages.

There are quantitative figures in Figure 39 and Figure 40. Both in Figure 39 and Figure 40 assume that there are 1024 group members, and random members leave a group in a same time. The results are average number from 2000 set random members.

Figure 39 shows the number of multicast messages sent by manager when members leave a group in a same time. IPTVP manager sends least number of multicast messages. LKH is twice bigger due to each node's corresponding administration key need to send to both leaves.

In Figure 40, IPTVP manager' computational costs are the least. OFT needs to generate blind keys, causing C_E+2C_f. computational costs. LKH encrypts the messages, but those messages need to transmit to both leaves.

Table 12: Efficiency analyses when members leave a group

	SKDC	**LKH**	**OFT**	**IPTVP**
Msg	n-l	2*sl*-l	*sl*	*sl*-*l*

S_Ccomp	$(n-l)C_E$	$C_E(2sl-l)$	$(C_E+2C_f)*sl$	$(sl-l)*C_E+(sl-l)*C_K$
Max. M_comp	C_E	$C_E(\log_2 n)$	$C_E+C_f(\log_2 n)$	$C_E+C_K+C_R+C_R(\log_2(n-1))$
Min. M_comp	C_E	C_E	C_E+C_f	C_K+C_R

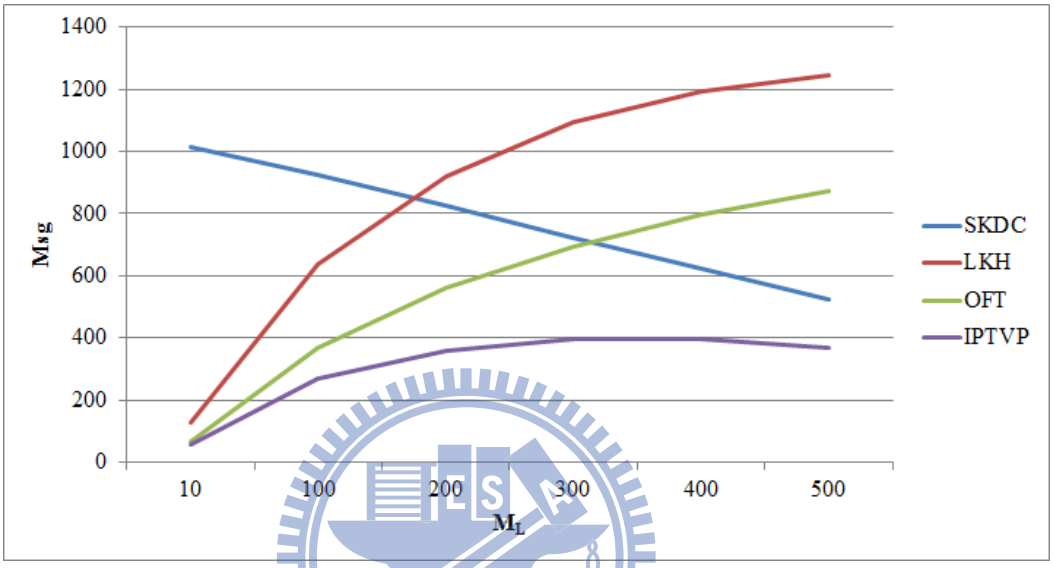


Figure 39: Number of rekeying messages when members leave a group

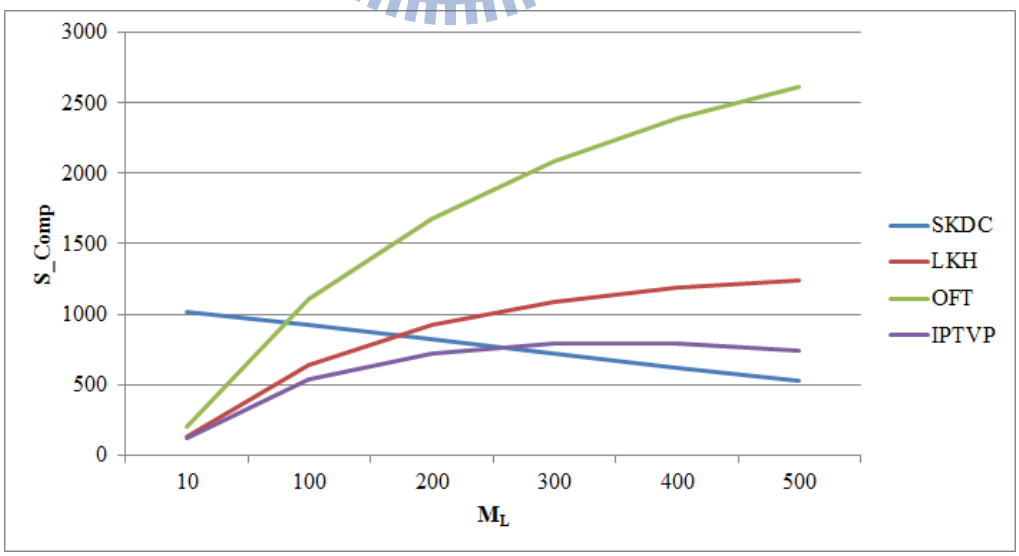


Figure 40: The computational costs of service manager when members leave a group

In Figure 39 and Figure 40, there are curve lines instead of straight lines. The reason is

some interior nodes' corresponding administration keys do not sent by managers. If an interior node' all leaves' members are leavers, manager do not send rekey messages to group members. However, SKDC don't use administration keys. Hence, SKDC shows a straight line in simulation result figures. Besides, IPTVP's computational costs are the least only when less than 250 group members leave a group, which the size of the group is 2^{10}. As shown in Figure 40.

4.2.1.5 Scenario 5: Storage analyses

In this scenario, there are only storage analyses between protocols. There are number of keys service manager needs to keep and each user needs to keep. Those keys have different functions and those are MPK, KEK, AK, RGK, and R values.

As shown in Table 13 in a quantitative way. IPTVP provider has to keep twice number of keys than OFT and LKH providers. SKDC provider only needs to keep same number of members with keys. Correspond with service manager's storages, each member's storages in IPTVP service are twice big than in OFT and LKH services, and SKDC service' each member only needs to keep 2 keys. The simulation results are Figure 42 and Figure 41. With group members increasing, the storages raise.

Table 13: Storage analyses

	SKDC	**LKH**	**OFT**	**IPTVP**
S_St.	N	2n	2n	4n
M_St.	2	$\log_2 n+1$	$\log_2 n+1$	$2(\log_2 n+1)$

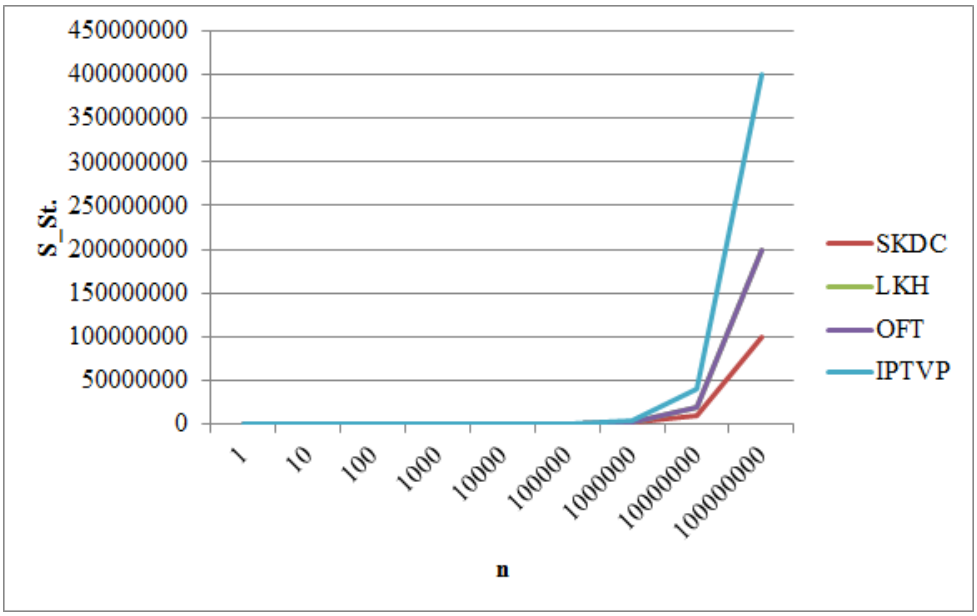


Figure 41: Service manager's storages

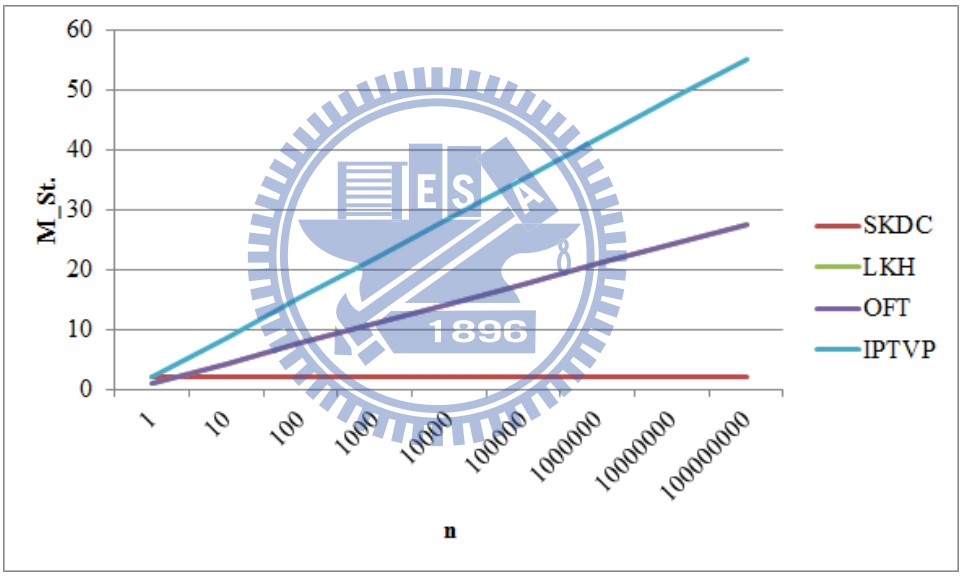


Figure 42: Each member's storages

4.2.2 Simulation Results: Tree Balance

In this section, there are showing the simulation results in section 3.5. Those results assume that the trees are binary trees and in sequence order. Besides, the numbers of joiner and leaver are random numbers. The two situations are showing in following examples.

4.2.2.1 The number of joiners is more than a tree's capacity.

In this example, assume a tree with 16 leaf nodes and the numbers 4, 10, 18 leaf nodes are vacant. If eight members join the group in the same time, the tree needs to rebuild.

The new tree is 32 leaf nodes with 11 vacant leaf nodes. Firstly the vacant leaf nodes are occupied. Then the rest joiners are added after new tree constructed, as described in 3.5.2. The results are Figure 43 and more clearly in Figure 44 and Figure 45.

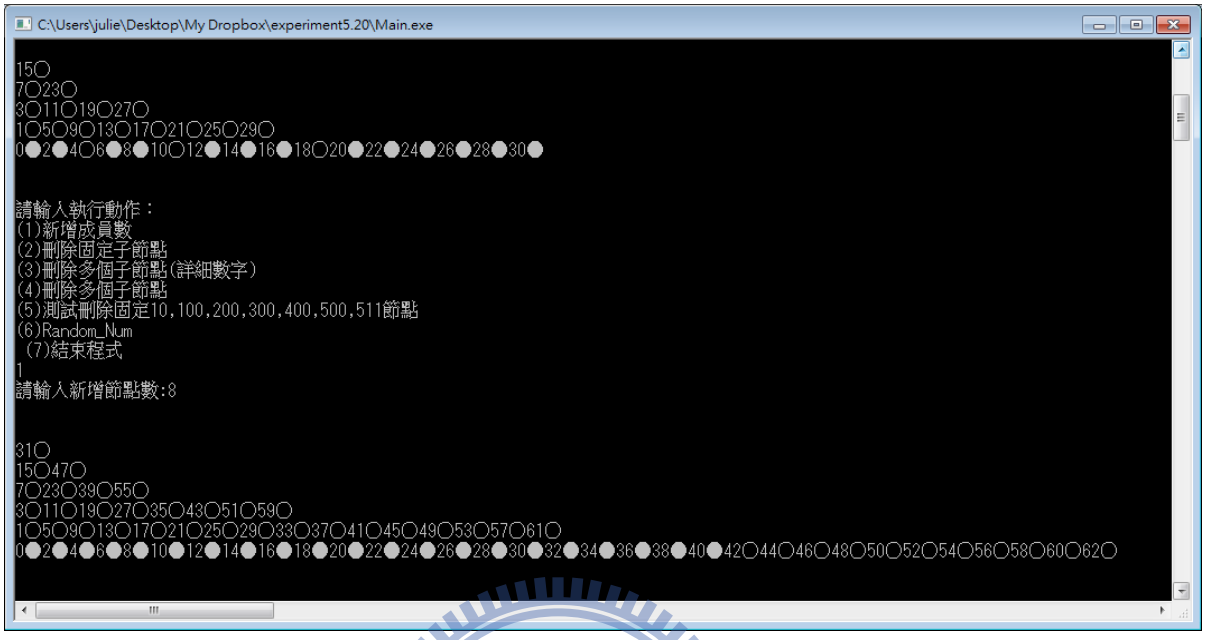


Figure 43: The number of joiners is more than a tree's capacity

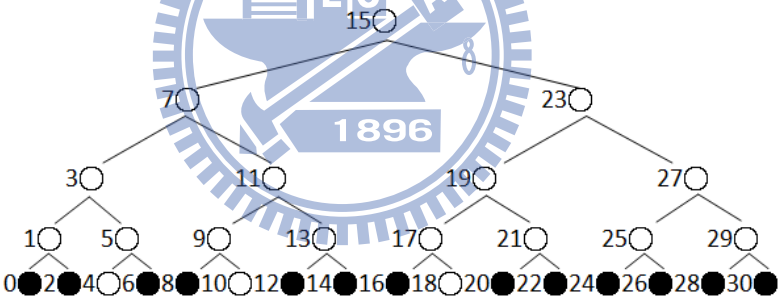


Figure 44: A tree is 16 leaf nodes with 3 vacant leaf nodes

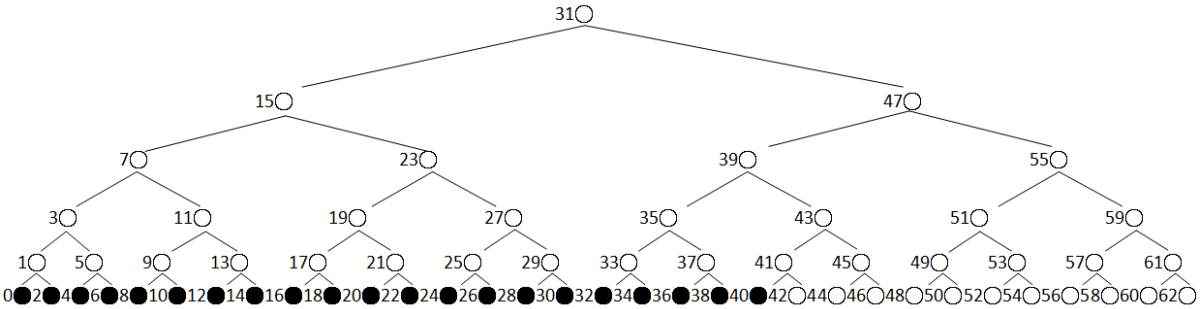


Figure 45: The new tree is 32 leaf nodes with 11 vacant leaf nodes

4.2.2.2 The number of leavers is more than a half of tree's capacity

Suppose there are a tree with 32 leaf nodes, and 32 group members. When 18

members leave the group at the same time, the tree needs to reconstruct. If numbers of those 18 members are 10, 26, 38, 8, 32, 54, 36, 42, 34, 58, 44, 2, 6, 12, 14, 48, 50, 40, the simulation results are Figure 46 and Figure 48. Figure 47 and Figure 49 illustrate the results more specifically.

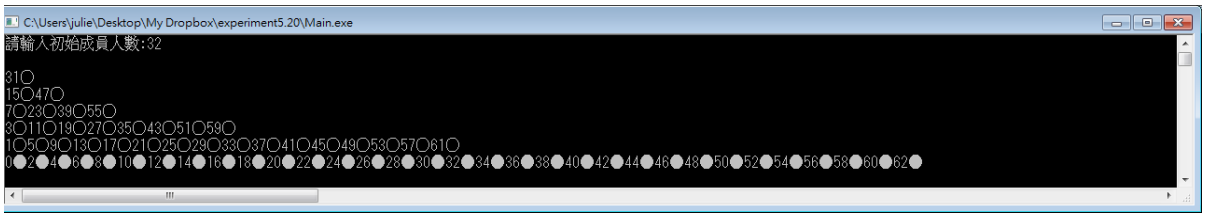


Figure 46: A tree with 32 leaf nodes, and 32 group members (1)

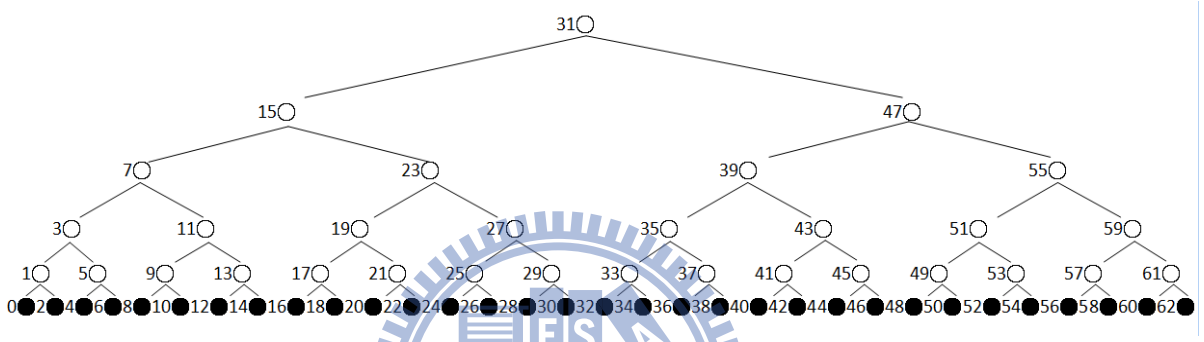


Figure 47: A tree with 32 leaf nodes, and 32 group members (2)

Figure 48: A new tree after reconstruction (1)

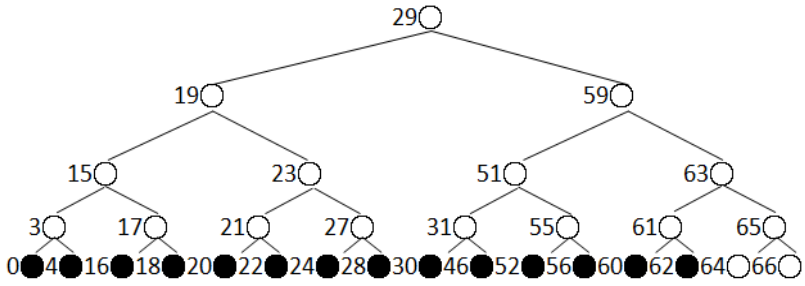


Figure 49: A new tree after reconstruction (2)

4.3 Discussions

In security analysis, there prevents collusion attacks and provide both forward and backward securities comparing with Sun's et al. CAS [17]. This thesis also provides stronger protocol comparing with improvement of Sun's et al. CAS [9]. In security requirements, this thesis provides same with other group protocols.

In analytical analysis, three indexes and five scenarios are used to analyze the efficiency of protocols. The three indexes are the computational costs, number of rekeying messages and storages. As described scenario 3 and scenario 4, the SKDC don't have administrative keys which is different with IPTVP, LKH and OFT. The computational costs and number of rekeying messages for SKDC's routinely updates are higher than LKH, OFT and IPTVP. Therefore, we only compare IPTVP with the average results from LKH and OFT summarizations.

In scenario 1 and 3, manager's computational costs and rekeying messages sent are only one broadcast plain text. IPTVP is better than other protocols at least 98% when member/members join a group, as shown in Figure 51, Figure 50, Figure 52 and Figure 53. When a member leaves a group, in scenario 2, IPTVP's service manager's computational costs are same with LKH, and the number of rekeying messages is same with OFT. IPTVP still is the least in three protocols. In scenario 4, IPTVP's both computational costs of service manager and rekeying messages are the least. IPTVP's numbers of rekeying messages are at least 40% less than other protocols, as shown in Figure 54. IPTVP's computational costs of service manager are at least 30% less than other protocols as shown in Figure 55. Even though, the storages are twice bigger comparing with OFT and LKH. This thesis provides a secure and efficient key management protocol. Finally, this thesis also completes the simulation experiment that keeping tree in balance.

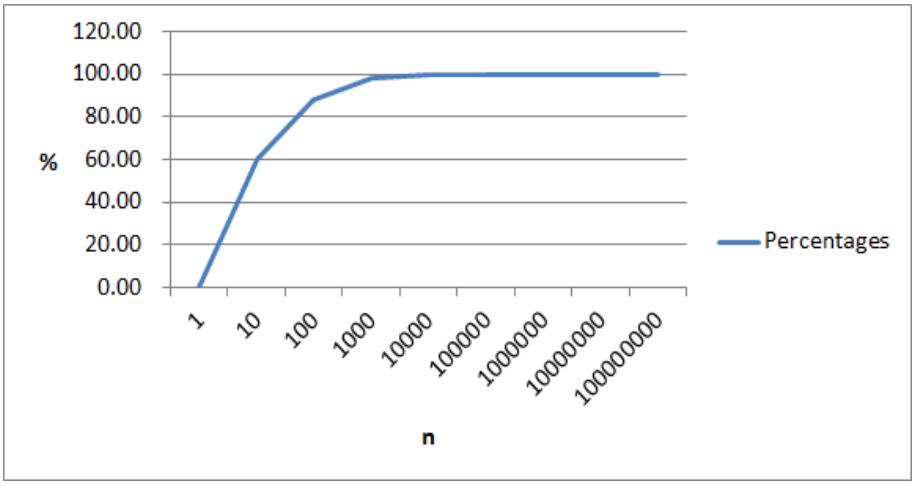


Figure 50: The percentages that IPTVP less than other protocols' number of rekeying messages in scenario 1

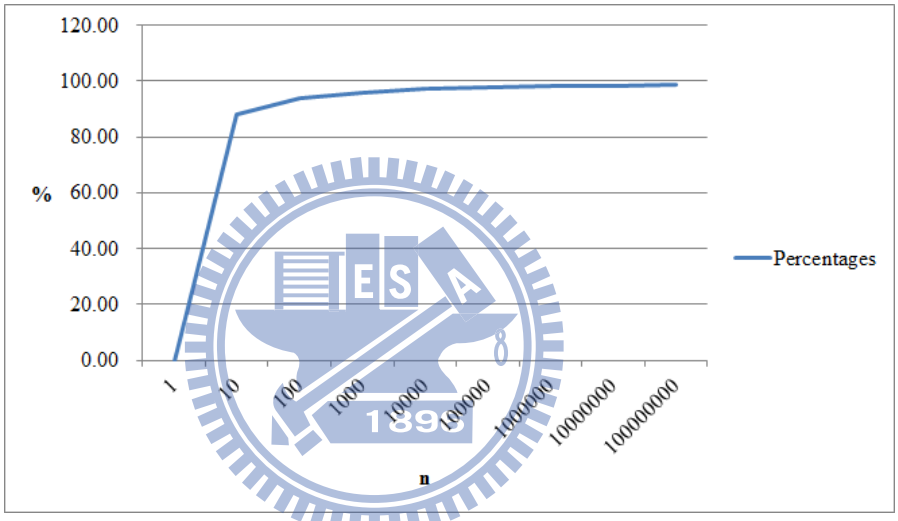


Figure 51: The percentages that IPTVP less than other protocols' computational costs in scenario 1

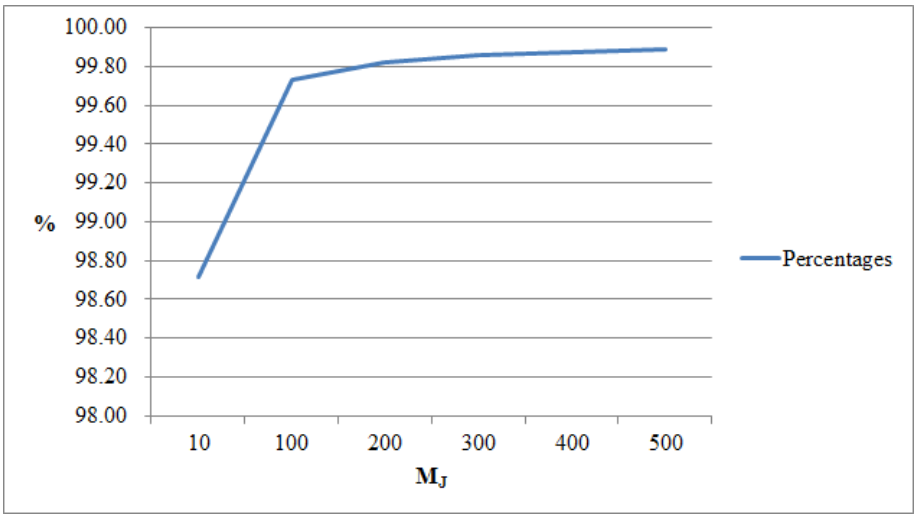


Figure 52: The percentages that IPTVP less than other protocols' number of rekeying messages in scenario 3

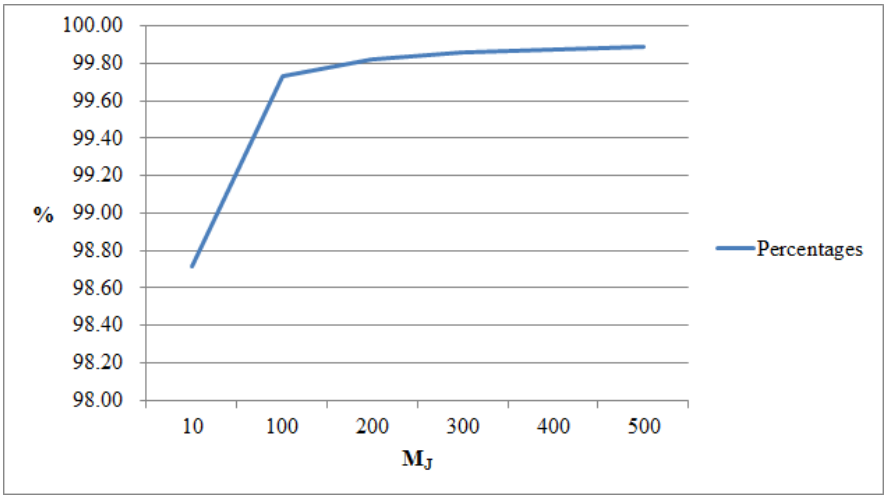


Figure 53: The percentages that IPTVP less than other protocols' computational costs in scenario 3

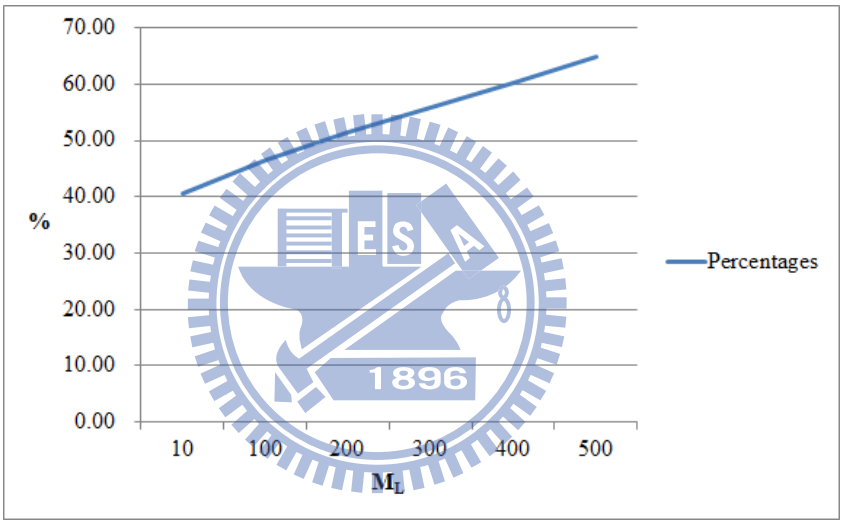


Figure 54: The percentages that IPTVP less than other protocols' number of rekeying messages in scenario 4

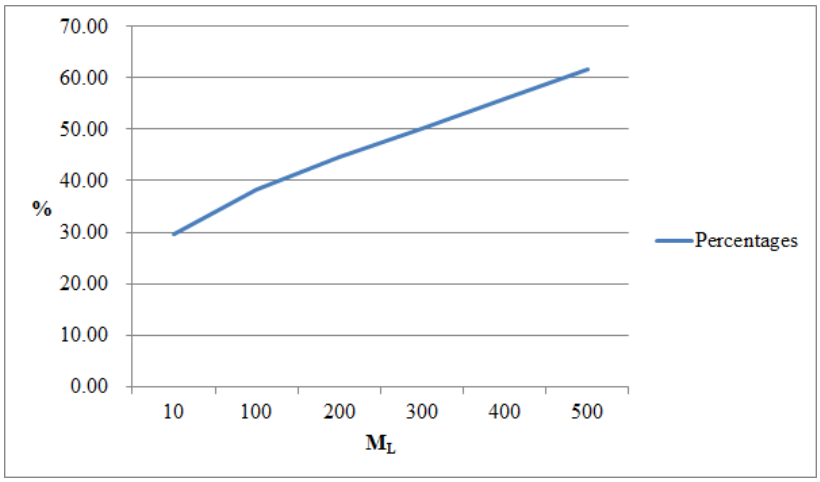


Figure 55: The percentages that IPTVP less than other protocols' computational costs in scenario 4

Chapter 5 Conclusions and Future Works

In this section, there are going to conclude and express the contributions in this thesis. Then, there are some suggestions to future research discussions.

5.1 Conclusions

In this thesis, we propose an idea of channel-based key management protocol for IPTV service. The proposed protocol helps the IPTV service provider easily implement keys' update and the number of rekeying messages to be sent. Following this thesis, the background, motivations, and objectives are described in chapter1. In chaper2, there describe the IPTV transmitting environment and infrastructures, different group key protocols, and secure requirement in multicasting group. Those issues illustrate the reason to adopt centralized group key management. Besides, there exist collusion attack secure problem and scale problem, which the number of transmitting messages and service's computation are large in Sun et al. CAS.

In chapter3, channel-based key management protocol for IPTV service is proposed. There are three phases: subscriber register phase, channel subscribing phase, and membership management phase. When a group membership is changed, a rekeying operation is used to inform group members automatically updating keys. Then, service provider generates and distributes the new group key for the group when Leave Operation or balance tree Operation is triggered. There are three operations to support rekeying requirement: Join Operation, Leave Operation, and Per_update Operation. And the tree balance operation maintaining efficient rekeying operation.

In this IPTVP's security analysis, there prevents collusion attacks and provide both forward and backward securities comparing with Sun's et al. CAS. Those security requirements also are provided same with other group protocols. In efficiency analysis, the

service manager's computational costs and rekeying messages sent obviously are cutting down to only one broadcast plain text when members join. This protocol is at least 98% better than other protocols. Those rekeying messages sent by service manager in IPTVP are same with OFT when a member leaves a group. IPTVP's service manager's computational costs are also same with LKH. This protocol still is the least in three protocols. IPTVP's both computational costs of service manager and rekeying messages are the least when members leave a group. IPTVP's numbers of rekeying messages are at least 40% less than other protocols. IPTVP's computational costs of service manager are at least 30% less than other protocols. Even though, the storages are multiplicative more comparing with OFT and LKH. Finally, this thesis also completes the simulation experiment that keeping tree in balance.

5.2 Future Works

This thesis supposes that the IPTV transmitting environment is matured, but there still are some challenges have to solve [16]. In simulation results and analyses, the storages are also a problem in this thesis, which are two times than other protocols. There are no analytical analyses in tree balance simulation, too. Besides, this thesis does not specifically express authorization secure problems and the rekeying processes in tree rebalance. To provide robust key management protocol, authentication, access control, denial-of-service attack (DOS), spyware, intrusion…etc. need to be concerned with.

References

[1] ATIS-IIF, http://www.atis.org/IIF/index.asp

[2] DVB, http://www.dvb.org/index.xml

[3] Functional requirements and architecture for IPTV security aspects, telecommunication standardization sector of ITU, Feb. 2009

[4] Suguru Higashino[1], Hideo Imanaka, Akira Takahashi, Yoshinori Goto, Shinji Ishii, and Masahito Kawamori, International Standardization of IPTV at ITU-T IPTV-GSI, NTT Technical Review , Vol. 8 No. 5 May 201

[5] IPTV Forum Japan. http://www.iptvforum.jp/en/

[6] IPTV functional architecture, telecommunication standardization sector of ITU, Sep. 2008

[7] IPTV Global Forecast –2009 to 2013 Semiannual IPTV Global Forecast Report, MRG, Inc.,Nov. 2009,pp. 109

[8] IPTV, http://en.wikipedia.org/wiki/IPTV

[9] Jung-Yoon Kim and Hyoung-Kee Choi,"Improvements on Sun et al.'s Conditional Access System in Pay-TV Broadcasting Systems", IEEE Trans. Multimedia, vol. 12, no. 4, pp. 337, Jun. 2010

[10] Chi-Chun Lo, Chun-Chieh Huang and Meng-Ju Lee, "A Channel-Based Key Management Protocol for IPTV Service", 8th annual IEEE Consumer Communications & Networking Conference, Jan. 2011

[11] J. Maisonneuve, M. Deschanel, J. Heiles, H. Liu, R. Sharpe, Y. Wu. An overview of IPTV standards Development, in: IEEE Transactions on Broadcasting vol.55, no3, June 2009 pp.315-328

[12] M. J. Moyer, J. R. Rao and P. Rohatgi, "A Survey of Security Issues in Multicast Communications," IEEE Network 13(6), Nov/Dec 1999, p.12-p.23.

[13] Open IPTV Forum. http://www.openiptvforum.org/

[14] S. Rafaeli and D. Hutchison, A Survey of Key Management for Secure Group Communication, ACM Computing Surveys, Vol. 35, No.3, September 2003, pp.309-329.

[1] NTT Cyber Solutions Laboratories, Yokosuka-shi, 239-0847 Japan

[15] A.T. Sherman and D.A. McGrew, Key Establishment in Large Dynamic Groups Using One-Way Function Trees, IEEE Transactions on Software Engineering, Vol.29, No.5, May 2003, pp.444-458.

[16] Y. Siao, X. Du, J. Zhang, F. Hu, and S. Guizani, "Internet Protocol Television (IPTV): The Killer Application for the Next-Generation Internet," IEEE Comm. Mag., pp. 126–134, Nov. 2007

[17] H. M. Sun, C. M. Chen, and C. Z. Shieh, "Flexible-Pay-Per-Channel: A New Model for Content Access Control in Pay-TV Broadcasting Systems," IEEE Trans. Multimedia, vol. 10, no. 6, pp. 1109-1120, Oct. 2008

[18] TISPAN, http://www.etsi.org/tispan/

[19] C.K. Wong, M. G. Gouda, and S.S. Lam, Secure Group Communications Using Key Graphs, IEEE/ACM Transactions on Networking, Vol.8, No.1, February 2000, pp.16-30.

[20] IPTV 新興商業模式與管理之研究期末報告,工業技術研究院,Dev. 2006

[21] 李振中, IPTV 相關技術探討, 廣播電視資料館, May 2007