Key Management for Secure Group Applications in Wireless Networks

Yiling Wang

A Thesis Submitted in Fulfillment of
The Requirements for
The Degree of Doctor of Philosophy

Faculty of Information Technology

Monash University

2008

Abstract

The advent of wireless networks has seen an associated emergence of group applications such as multimedia teleconferencing, stock quoting and distance education. In order to secure group applications and prevent unauthorized users from accessing communication data that cannot be protected by wireless networks and IP multicast alone, group communication content needs to be encrypted by a shared group key. Key management is necessary to ensure the safety of this group key and to protect the group communication. However, problems arise when group key management schemes are applied to the wireless environment.

These problems relate to three issues: performance, security and network compatibility. Performance issues arise from the resource limitations of both wireless networks and mobile devices and related to problems such as high transmission error rates, limited bandwidths and insufficient computation power. These limitations restrict the operation of wireless group key management approaches. Operational efficiency is thus a high priority for wireless group key management. The second issue relates to security. Wireless group key management needs to have the capacity to ensure the safety of the group key in adverse circumstances such as collusion attack. The third issue relates to network compatibility. Each type of wireless network has its own network specific features that can affect the design and

operation of a group key management system. One wireless group key management approach is unlikely to be compatible with all wireless networks. A wireless group key management approach must therefore be network compatible. In this research, we focus our attention on the cellular wireless network since it is the dominant wireless network architecture and widely deployed around the world.

The purpose of this thesis is to propose approaches to enable efficient, secure and practical group key management in the cellular wireless network. The research conducted in this thesis focuses on three different levels: *the formal model, the system component level and the system solution level*.

At the formal model level, we identify the basic and essential components of a wireless group key management system. These components are the building blocks necessary to design a wireless group key management system. In addition, we identify the relationships between the issues of wireless group key management (performance, security and network compatibility) and these components. This identification helps to suggest methods to address these issues. Then, in order to analyze and evaluate the proposed wireless group key management approaches, a set of assessment parameters is proposed. Both the proposed wireless group key management system model and assessment parameters can be considered a guideline for system designers and implementers to design, analyze and evaluate wireless group key management approaches.

At the system component level, we propose three group key management approaches, each of them especially designed to tackle a particular problem in wireless group key management. At this level, a wireless group key management

architecture for the cellular wireless network is proposed. In this architecture, the group key management infrastructure seamlessly integrates with the underlying cellular wireless network structure to utilize the capacity of the wireless network to facilitate group key management. In order to tackle two particular wireless group key management problems, operational efficiency and multiple-membership changes, we develop two group key management approaches. To enable the efficient operation of key management in the cellular wireless network, a hybrid group key management approach - which operates within the cell of the cellular wireless network - is proposed. By performing micro-key management, this approach can reduce the operational costs associated with key management and improve the operational efficiency of wireless group key management. We also propose a group key management scheme - membership-oriented key management - to tackle the performance problem of multiple-membership changes. This approach, compared to traditional application-oriented group key management approaches, offers more effective management of multiple-membership changes.

Finally, at the system solution level, all three proposed approaches are integrated into a comprehensive wireless group key management solution for the cellular wireless network. This solution is tested against our assessment parameters to demonstrate that the solution offers an efficient, secure and practical key management for secure wireless group applications.

Acknowledgements

This thesis would not have been possible without the invaluable guidance, support, continual encouragement, discussion and advice from my two supervisors, Professor Bala Srinivasan and Dr Phu Dung Le. I would like to first and foremost express my deeply appreciation to them for their care, both of my study and personal life. Without their supervision, I would not have been able to achieve my aspirations.

I would like to dedicate this thesis to my parents to whom I owe my every success in the life. I thank my parents for their continuous support, care and encouragement. They have always been supportive of me in everything I have decided to do. Whenever I needed help, their suggestions and guidance always helped me to sort things out easily.

I owe special thanks to my nice officemate and friend, Huy Ngo, for many hours of pleasant conversations, research discussions and the sharing of perspectives on life. I also thank Xianping Wu and Osama Dandash for the research discussions. I have special gratitude toward my housemates Yen-Chien Lee and Isabel Chan. They helped me get through many things.

I finally wish to express my thanks to everyone at the Caulfield School of the Faculty of Information Technology for their support for everything including the scholarship provided during my PhD candidature.

Declaration

I hereby declare that this thesis contains no material which has been accepted for the award of any other degree or diploma at any university or equivalent institution and that, to the best of my knowledge and belief, this thesis contains no material previously published or written by another person, expect where due reference is made in the text of the thesis.

Yiling Wang

November 26, 2008

Publications

Wang, Y., & Le, P. D. (2005). *Secure Group Communications in Wireless Networks.* In proceedings of the Third International Conference on Advances in Mobile Multimedia (MoMM2005), pp. 241-252.

Wang, Y., & Le, P. D. (2005). *Scalable Multi-Subgroup Key Management in Wireless Networks.* International Journal of Computer Science and Network Security, Vol. 5(11), pp. 95-105.

Wang, Y., Damodaran, D., & Le, P. D. (2006). *Efficient Group Key Management in Wireless Networks.* In Proceedings of the Third International Conference on Information Technology: New Generations (ITNG 2006), pp. 432-437.

Wang, Y., Le, P. D., & Srinivasan, B. (2007). *Hybrid Group Key Management Scheme for Secure Wireless Multicast.* In Proceedings of the 6th IEEE International Conference on Computer and Information Science (ICIS 2007), pp. 346-351.

Wang, Y. & Le P. D. (2007). *Secure Group Communications in Wireless Networks*. Encyclopedia of Mobile Computing & Commerce Vol. 1, pp: 227-232, IGI Publishing.

Wang, Y. & Le P. D. (2007). *Efficient and Scalable Group Key Management in Wireless Networks.* Encyclopedia of Mobile Computing & Commerce Vol.2, pp: 832-838, IGI Publishing.

Chen, Y., Wang, Y., Wu, X., & Le, P. D. (2006). *The Design of Cluster-based Group Key Management System in Wireless Networks.* In Proceedings of the International Conference on Communication Technology (ICCT 2006), pp. 209-212.

Dandash, O., Wang, Y., Le, P. D., & Srinivasan, B. (2007). *A New Dynamic Key Generation Scheme for Fraudulent Internet Payment Prevention.* In Proceedings of the Fourth International Conference on Information Technology (ITNG '07), pp. 83-88.

Dandash, O., Wang, Y., Le, P. D., & Srinivasan, B. (2007). *A New Group Key Management Structure for Fraudulent Internet Banking Payments Detection.* In Proceedings of the 9th International Conference on Enterprise Information Systems, pp. 57-62.

Dandash, O., Wang, Y., Le, P. D., & Srinivasan, B. (2008). *Fraudulent Internet Banking Payments Prevention Using Dynamic Key*. Journal of Networks Vol. 3(1), pp. 25-34.

Contents

List of Figures

List of Tables

Chapter 1

Introduction

1.1 Group Communications in Wireless Networks

In recent years, computing and networking have shifted from a static wired network model to the “anytime, anywhere” mobile service model. The latest advancements in wireless technologies not only facilitate such pervasive computing but also provide the ubiquitous communication coverage that is coveted by mobile device providers. The future wireless networks will gradually become the primary interface for network communication and main platform of applications and services [Salkintzis, 2004].

In a wireless environment, a number of applications are inherently group-oriented at application level. In addition, wireless group applications cater for a wide variety of domains, including mobile commerce (m-commerce), military command and control, distance education and intelligent transportation systems [Gossain & Cordeiro et al., 2002; Varshney, 2002]. In m-commerce, many new applications (including mobile marketing, mobile sales and mobile auctions) gain

significant benefits from the group communication model, because the number of potential customers is greatly increased by commercial information being efficiently delivered to a large number of receivers. In a shopping center, customers with wireless appliances could receive messages advertising sales as they enter shopping precinct. To gain efficiencies and marketing effectiveness, group, rather than individuals, would be targeted. In military environments, tactical information can be broadcast to soldiers, tanks and planes simultaneously to improve their cooperative capacity. Distance education and entertainment services can be simultaneously provided to mobile users. In intelligent transportation management systems, the latest traffic information, including the most direct and least time-consuming routes can be efficiently delivered to nearby drivers who subscribe to such services via a group communication model.

Wireless group communication is not only driven by the applications; it is also motivated by communication content. Multimedia content has been growing at an exponential rate on the Internet due to the popularity of digital devices. The combination of content and communication technologies has created opportunities for businesses to meet the growing global demand for information and entertainment. Applications and services such as IPTV [O'Driscoll, 2008], video-on-demand and video conferencing have brought the Internet into the multimedia era. However, multimedia applications are bandwidth-intensive. They require large bandwidth to transmit multimedia content even if the communication is already compressed. Bandwidth is the biggest obstacle when transmitting multimedia content in wireless networks, because the bandwidth is quite narrow compared to that of wired networks.

It is inefficient to apply a point-to-point communication model to distribute multimedia content to hundreds and thousands of users simultaneously. In order to reduce the workload of server and wireless network for multimedia communication, the solution is to efficiently apply a group communication model.

In summary, group communication represents an important model of application traffic in wireless networks. Moreover, group communication model is a necessity for the emerging multimedia applications and services in wireless networks.

1.2 Access Control in Wireless Group Communication

A number of approaches can be considered when implementing a group communication model. These include broadcasting, overlay multicasting at the application level [Banerjee & Bhattacharjee et al., 2002; Hosseini & Ahmed et al., 2007; Zhang & Jamin et al., 2002] and IP multicasting [Deering, 1988, 1989; Goncalves & Niles, 1999]. Of these, IP multicast transmission is considered to be the most efficient and suitable; it is also the widest deployed underlying transmission mechanism to implement group communication due to its bandwidth conservation technology [Holbrook & Cheriton, 1999; Zhang & Jamin et al., 2002]. In IP multicast transmission, instead of sending a separate copy of the packet to each individual receiver, a sender only transmits one copy of the packet to a group address (Class D IP address from 224.0.0.0 to 239.255.255.255) [IANA, 2008; Zappala & Lo et al., 2004]. It is the responsibility of network and multicast-enabled routers to make copies of the packets on their way from the sender to multiple receivers located at

different segments of the network. IP multicast transmission is illustrated in Figure 1.1.

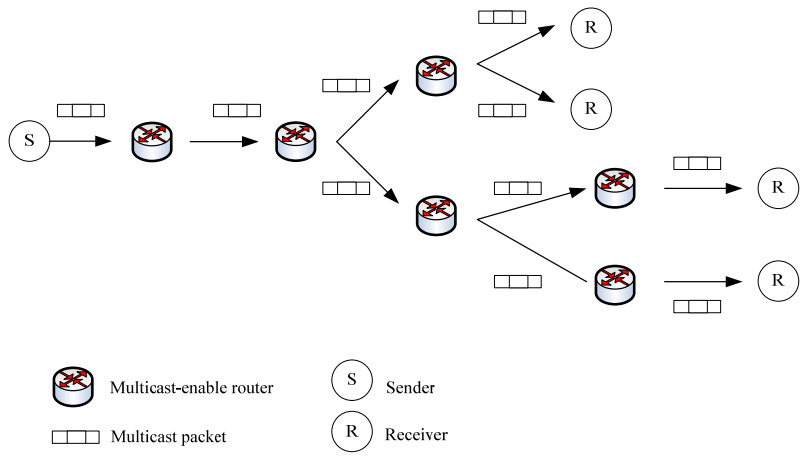


Figure 1.1 IP multicast transmission mechanism

IP multicast transmission provides good scalability when implementing group communication due to its open structure. Receivers can join a group and a sender can transmit data to a group without any interaction with a central entity. However, this open model lacks any security measures that would enforce access control and protect group communication. In an IP multicast application, any receiver can request data and a receiver does not need to directly contact sender(s) to express its interest in receiving data. Instead, a receiver sends a message to the first multicast-enabled router to register that it is interested in receiving data for a given group. Due to this anonymous receiving model, the sender is unable to enforce any access control to manage group membership. In some cases, this limitation would render this model unsuitable for commercial situations, as a service provider may prefer to limit content distribution to subscribers who pay for the service. When IP

multicast applications operate in wireless networks, it is more difficult to enforce access control due to the broadcasting nature of wireless networks. Once the group communication data flows into the wireless network, all hosts within the scope of the wireless network can access the data – whether they are members or not, or whether all members pay for the service or just one does. As a result, IP multicast transmission is unable to offer any measures to protect group communication content in wireless networks. Hence, access control needs to be enforced in wireless networks to ensure the security of group communication. This can be achieved through key management.

1.3 Key Management in Wireless Group Communication

In a wireless environment, access control is the most fundamental and critical security issue in group communication [Gong & Shacham, 1995; Judge & Ammar, 2003; McHugh & Michale, 1999]. Generally, access control can be achieved by applying encryption. A shared key, called group key or traffic encryption key (TEK), is used to encrypt the group communication data and is distributed to all legitimate group members. Only the members who own this group key can access the communication content. The confidentiality and integrity of the group's communication rely on the safety of the group key. Management of the group key thus plays an important role in the security of group communication [Bruschi & Rosti, 2002; Hardjono & Tsudik, 1999; Kruus, 1998; Moyer & Rao et al., 1999; Waldvogel & Caronni et al., 1999].

Key management in group communication is very different from that in the point-to-point communication model. In the point-to-point model, the encryption key can be generated by negotiation through protocols such as the Diffie-Hellman key exchange protocol [Diffie & Hellman, 1976] or it can be generated by one side and then sent to another side. When one side leaves the communication, the connection is automatically terminated, and the encryption key is discarded. Consequently there is no need to update the encryption key. However, in group communication, a group may have many receivers, and the efficient generation, regeneration and distribution of the group key to all receivers is a complicated and challenging task. When one or several members leave the group, the group communication is still active and no one can force the departing member to forget the key. In order to prevent ex-group members from accessing future communication data, the group key needs to be updated. The group key also needs to be updated whenever a new user joins the group. A new joining member might record the encrypted group communication before it joins the group. In order to decrypt these recorded data, the user joins the group for a short time to obtain the group key. If the group key is not updated, the user could access group content to which it is not entitled. Furthermore, keys for encrypting data should be changed periodically. Cryptographers frown on encrypting a lot of data with the same key because the data is susceptible to the cryptanalysis attack. To sum up, the major task of group key management is to generate, distribute and update the group key to protect the security of group communication.

Wireless networks impose limitations on group key management when the key management tasks are performed in a wireless environment. The limitations of wireless networks can be classified into two categories: resource limitations of mobile devices and characteristics of wireless networks. Mobile devices have less computation power, less storage capacity and a limited power supply compared to desktop computers. These restrictions prevent mobile devices from performing complicated security measures such as public key algorithms. Hence, group key management approaches may not be adopted by wireless networks because algorithmic techniques are time-consuming and computationally complicated. On the other hand, the characteristics of wireless networks - users' mobility, narrow bandwidths and a high transmission error rates - also place restrictions on group key management. Although mobility is a property unique to wireless networks, it nonetheless poses a new challenge to group key management: how to efficiently deliver an encryption key to mobile users when users move from one location to another. The small bandwidth and high error rates associated with wireless networks also require group key management solutions to specifically address these issues. Moreover, there are several types of wireless networks currently in existence such as wireless LAN, cellular wireless network, and wireless sensor network [Pahlavan & Krishnamurthy, 2001]. Each of these has its own network-specific features in terms of communication capacity, computation power and network infrastructure. These diverse network-specific features can affect the design and operation of wireless group key management approaches. Hence, a group key management solution for one kind of wireless environment might not be suitable for another environment.

These issues considerably increase the complexity of group key management in wireless networks.

In summary, the major problems and consequent challenges of group key management in the wireless environment are:

- Performance problems

  Group key management in wireless networks needs to provide operational efficiency in communication, computation and key storage to overcome the restrictions of both the wireless network and mobile devices.

- Security problems

  Group key management in wireless networks needs to implement measures to protect the safety of the group key when users join or leave the group. Moreover wireless group key management is required to offer security measures to protect the group key and other supporting keys from being compromised by non-group users or ex-group members.

- Network-compatible problems

  Due to the variety of wireless networks, there is no single group key management approach compatible with all wireless networks. Therefore, a group key management solution needs to be wireless network compatible.

1.4 Motivations and Objectives of the Thesis

Group key management plays a critical role in enforcing access control to secure group applications in the wireless environment. However, the limited resources of both wireless networks and mobile devices impose restrictions on the design and operation of group key management approaches. Currently, wireless group key management encounters problems relating to performance, security and network compatibility. These problems motivated us to conduct the research in this thesis to solve the problems of wireless group key management and to accelerate the successful deployment of group applications in wireless networks.

The primary objectives of this thesis are:

- To develop a formal group key management model for wireless networks. This model is developed to identify the fundamental and essential components in wireless group key management system. These critical components can be considered as building blocks for system designers to develop wireless group key management systems.
- To develop a group key management solution for the cellular wireless network. This solution is designed to provide efficient, secure and practical group key management in the wireless environment with the following two components.
  - Group key management architecture for the cellular wireless network.

    The purpose of this architecture is to integrate the group key management structure with the underlying cellular wireless network topology to facilitate the operation of key management.

- Group key management approaches (algorithms).

  The purpose of developing special group key management approaches is to efficiently perform key management based on the proposed group key management architecture.

1.5 Contribution of the Thesis

The major contributions of this thesis are as follows.

- To develop a formal group key management model for wireless networks.

  In this model, the important and essential components in wireless group key management system and the relationships between them are identified. These important components are the building blocks to design wireless group key management systems. Based on this model, the links between the problems of wireless group key management and these key components are also identified, providing possible methods to address these problems. In order to analyze and evaluate wireless group key management systems, a set of assessment parameters is developed. The proposed model and assessment parameters can be seen as guidelines for the design and evaluation of wireless group key management systems.

- To design a group key management architecture for the cellular wireless network.

  This wireless group key management architecture is developed to seamlessly integrate the key management structure with the underlying cellular wireless network's infrastructure. This utilizes the capacity of the wireless network to

facilitate the operation of group key management.

- To develop and verify two group key management approaches (algorithms).

  These two group key management approaches are designed to tackle the particular performance problems in wireless group key management. One group key management approach is tailored to be operated within the wireless cell to reduce the operational costs and to improve the performance of key management in the cellular wireless network. The other algorithm is especially designed to reorganize the key management structure to address the multiple-membership changes issue in group key management.

- To develop and verify a group key management solution for the cellular wireless network.

  This solution is developed by integrating the proposed wireless group key management architecture and two group key management approaches. This integrated solution satisfies all the requirements of performance and security for wireless group key management systems and so can be considered as an efficient and secure group key management solution for possible deployment in the cellular wireless network.

- To formally formulize the users' actions of join and leave in group communication.

  The users' actions of join and leave are formally formulized from the view of statistics. This formulization provides a powerful tool for system designers to quantitatively analyze and evaluate the capacity of a wireless group key management system based on users' behavior.

1.6 Thesis Organization

The thesis is organized as follows.

Chapter 1 introduces and explains the need for, and problems associated with group key management. The motivation for this thesis is described, along with its objectives and practical and theoretical contributions. The chapter closes with a description of the thesis organization.

Chapter 2 reviews the existing group key management approaches. The advantages and disadvantages of these existing group key management schemes are investigated in detail in this chapter.

Chapter 3 presents a formal model for wireless group key management systems. In order to analyze and evaluate wireless group key management systems, a set of assessment parameters is also proposed. In addition, a wireless group key management architecture is developed in this chapter by applying the proposed model to the cellular wireless network.

Chapter 4 presents a new hybrid group key management approach. The structure and operation of this scheme are investigated in depth in this chapter. The operational costs of this approach are analyzed and evaluated.

Chapter 5 discusses another novel group key management approach - membership-oriented key management - that is designed to address a serious performance issue with respect to multiple-membership changes.

Chapter 6 presents a comprehensive group key management solution for the cellular wireless network. In this chapter, the structure and operation of this solution

are investigated through a case study. The solution is analyzed and evaluated based on the assessment parameters proposed in Chapter 3.

Chapter 7 summarizes the achievements of this thesis and highlights its contributions. Possible future research is also discussed in this chapter.

Chapter 2

Group Key Management Schemes

Data communication driven by IP multicasting provides an efficient transmission model for group applications and services. However, due to the open structure of IP multicast transmission, this approach lacks access controls to protect the communication content. In wireless networks that are a shared medium-based environment, any user in the network can access the wireless communication contents. The only way to enforce access control in wireless group communication is through encryption. A shared key is applied to encrypt the communication to prevent unauthorized users accessing the content. This encryption key is called a group key or a traffic encryption key (TEK), and is shared among all legitimate group members. The security of group communication is fully dependent on the safety of this group key.

Group key management plays an important role in achieving secure group communication. The tasks of group key management are to provide: (i) member identification and authentication, (ii) access control and (iii) management of keying

material including the group key and all the supporting keys [Rafaeli & Hutchison, 2003; Zou & Ramamurthy et al., 2005]. In this thesis, we focus on the last task, because the most important issue in group key management is to ensure the safety of keying materials [Hardjono & Tsudik, 1999; Kruus, 1998]. There are three missions being carried out by group key management.

- Key generation

  Key generation refers to the generation of the group key and all other supporting keys that help key distribution controller to distribute the group key to all legitimate receivers.

- Key distribution

  Key distribution pertains to the efficient, secure and reliable delivery of keying materials to group members. Because group members may be geographically dispersed or move from one location to another in wireless networks, the efficient delivery of the group key to all legitimate members is the most important task in group key management.

- Key updating (rekeying)

  Key updating refers to the process of changing the group key and supporting keys and sending the updated keys to group members. The group key needs to be updated when membership changes, such as join and leave, are enacted. The purpose of key updating is to enforce backward and forward secrecy [Hardjono & Dondeti, 2003; Kim & Perrig et al., 2004b]. When a user joins a group, in order to prevent the user from accessing communication content that has existed before it joins, the group key needs to be changed. This refers to backward

secrecy. On the other hand, when a member leaves a group, the group key also needs to be updated in order to prevent the departing member from accessing future group communication. This is called forward secrecy.

In order to promote efficient and secure group key management, substantial research work has been carried out in the group key management area over the last decade. A number of group key management schemes have been proposed in the literature to address the problems and challenges of group key management. These include approaches such as Scalable Multicast Key Distribution [Ballardie, 1996], Group Key Management Protocol (GKMP) [H. Harney & Muckenhirn, 1997a, 1997b], MARKS [Briscoe, 1999], Logical Key Hierarchy (LKH) [Wong & Gouda et al., 1998, 2000], Efficient Large-Group Key Distribution (ELK) [Perrig & Song et al., 2001], Kronos [Setia & Koussih et al., 2000], Distributed Logical Key Hierarchy [Rodeh & Birman et al., 2000], Two-level Rekeying Architecture [DeCleene & Dondeti et al., 2001], Topology-Marching Key Management [Sun & Trappe et al., 2002, 2003], One-way Function Tree (OFT) [Sherman & McGrew, 2003], SAKM [Challal & Bettahar et al., 2004], Contributory Key Agreement [Amir & Kim et al., 2004], and last hop topology sensitive multicasting key management [Ghosh & Anjum, 2005].

These schemes can be classified into two main categories: network-independent group key management and network-dependent wireless group key management. The principal difference between them is whether the group key management approach takes into account the features of the underlying network architecture when it

performs key management. The network-independent group key management approaches do not consider the underlying network infrastructure. Network-independent approaches can be applied in both wired and wireless networks, and can be further classified into three types:

- centralized group key management scheme;
- decentralized architecture; and
- distributed key management schemes.

In contrast, the network-dependent wireless group key management approach intends to utilize the capacity of the underlying wireless network to facilitate group key management in the wireless environment.

The classification of group key management is shown in Figure 2.1.

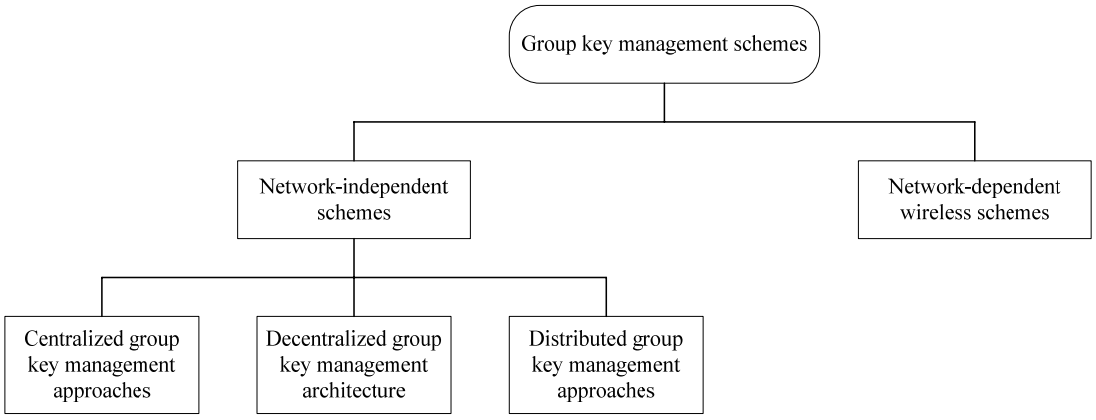


Figure 2.1 Classification of group key management approaches

In this chapter, we investigate these four types of group key management approaches. For each type, the most prominent and important group key management schemes in the course of research and development of group key management are reviewed.

The remainder of this chapter is organized as follows. In section 2.1, we investigate a formal group key management model provided by the Internet Engineering Task Force (IETF). In section 2.2, we introduce the centralized group key management schemes. We discuss the decentralized architecture and distributed group key management schemes in section 2.3 and 2.4 respectively. In section 2.5, we discuss the performance of network-independent approaches in wireless networks. We illustrate the network-dependent wireless group key management approach in section 2.6. A chapter summary is provided in section 2.7.

2.1 IETF Group Key Management Model

The IETF group key management model [Hardjono & Baugher et al., 2001] has proposed as a guildline for the designs of group key management system. The IETF model (Figure 2.2) includes both centralized and distributed group key management., and assumes that a one-to-many multicast is applied, so there is only one single sender. The entities involved in the group key management are key distributors (KDs), a member sender and member receivers. In Figure 2.2, each member is associated with one KD via a key management channel (denoted by a dotted double-arrow line). Members receive the keying materials via this channel. The member sender transmits encrypted data packets to the multicast group (denoted by thick full lines), which is received by both the local receivers and the remote receivers through a multicast distribution tree. In the distributed model, KDs synchronize the group key via the key management channel as well.

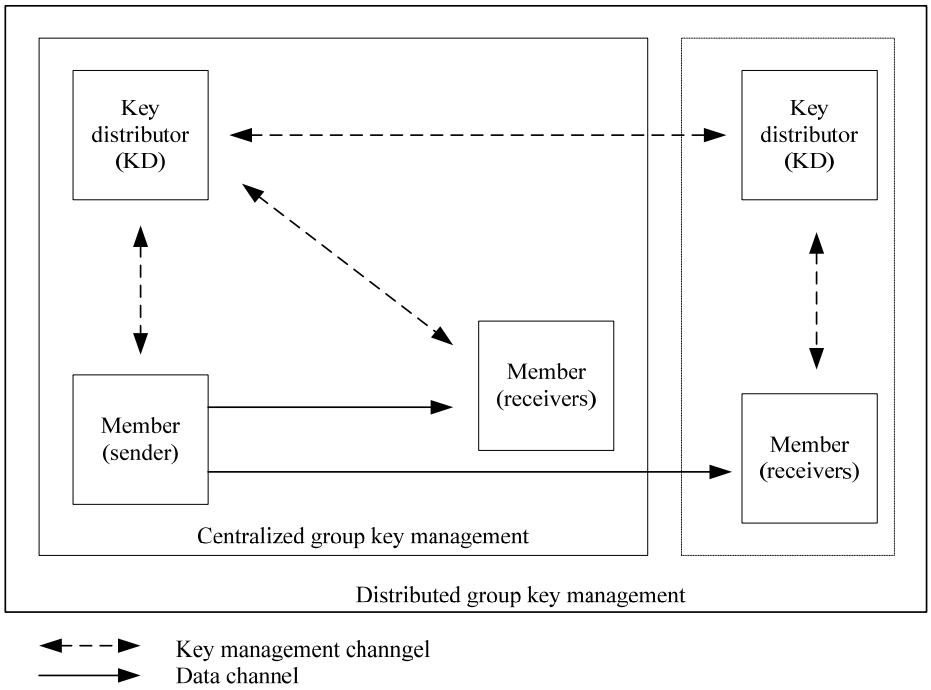


Figure 2.2 IETF model of group key management

The contribution of the IETF model is that it identifies the entities involved in group key management and defines the information flow of key management among these entities. These definitions are able to facilitate the design of group key management systems. However, the following important issues are not discussed in this model:

- The role and function of the underlying network environment in group key management.

  Network environment refers to the topology and configuration of network entities such as routers, switches and specific servers. This environment provides the basic communication services to group applications. Therefore, it also has an impact on the design of group key management systems. For example, key distributors (KD) should be located as closely as possible to group members to

improve operational efficiency. In the wireless environment, due to the limited capacity of mobile devices, group key management systems need to cooperate closely with the underlying wireless network to fully utilize its capacity to facilitate key management operation. The network environment therefore plays an important role in wireless group key management and should be addressed in the design model. However, the IETF model is a network-independent model. It does not describe the function of the underlying network infrastructure and its relationship with other key management entities. Therefore, this model is not suitable for wireless networks.

- This model does not provide parameters to analyze and evaluate group key management systems.

  Designers of group key management system need to know the parameters that can be applied to analyze and evaluate group key management systems. However, the IETF model fails to define these parameters.

Due to the limitations of the IETF model, it does not meet the design requirements of group key management in wireless networks.

2.2 Centralized Group Key Management Schemes

In centralized group key management schemes, a single trusted entity called a key distribution center (KDC) is employed to manage the group key and other supporting keys for the whole group. Under the simplest centralized group key management scheme, the KDC assigns a secret key to each member in the group and

encrypts the group key separately with each member's secret key when rekeying. The cost of this rekeying scheme is proportionally linear to the group size, creating inefficient communication. In order to increase key updating efficiency, a hierarchical structure (called a key tree) [Wallner & Harder et al., 1999] has been introduced to minimize the utilization of communication bandwidth, computation power and key storage requirements for both members and key servers. Of the centralized group key management approaches, the Logical Key Hierarchy (LKH) [Caronni & Waldvogel et al., 1998; Wallner & Harder et al., 1999; Wong & Gouda et al., 1998, 2000] and the One-way Function Tree (OFT) [Sherman & McGrew, 2003] are the best-known and utilized schemes. In the next sections, these two approaches are discussed in depth.

2.2.1 Logical Key Hierarchy (LKH)

The Logical Key Hierarchy (LKH) approach has been proposed independently by Wallner et al. and Wong et al. [Wallner & Harder et al., 1999; Wong & Gouda et al., 1998, 2000] is one of the most prominent and efficient group key management algorithms. The major contribution of LKH is to apply a hierarchical structure, as shown in Figure 2.3, to facilitate group key management. Theoretically, any hierarchical tree structure can be applied in LKH. Without loss of generality, we apply a binary tree as an example to illustrate LKH, due to the simple establishment, management and node operations of a binary tree.

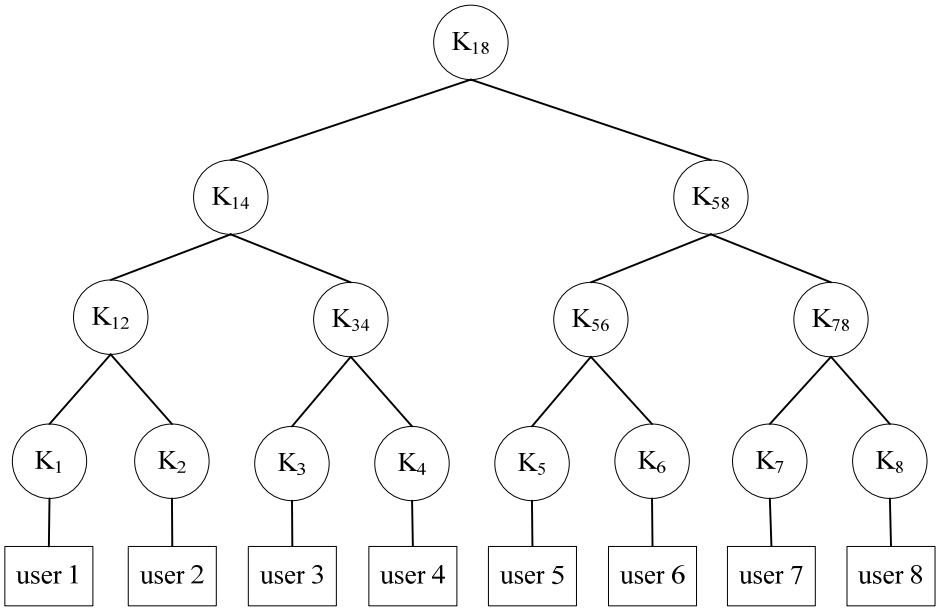


Figure 2.3 A LKH key tree

In a LKH key tree, the internal nodes of the tree hold supporting keys (key encryption keys (KEK)), which are used to encrypt the group key for distribution to group members. Each leaf node is associated with a group member and holds a pair-wise KEK (see keys ($k_1 - k_8$) in Figure 2.3). The KEK of leaf node is determined when a user joins the group and is only known to the joining member and the KDC. Each member needs to keep a set of KEKs along the path from its leaf node to the root node. For a balanced binary tree, each member stores $h+1$ keys, where h is the height of the tree and equals $\log_2 n$, and n is the number of group members. For example, in Figure 2.3, user 1 needs to know a set of keys $(k_1, k_{12}, k_{14}, k_{18})$. In the following sections, we investigate the key management when a user joins or leaves a group.

2.2.1.1 The Join Operation

When a new user joins a group, the KDC needs to add the user into its key tree. There are two different possible scenarios. In the first scenario, the KDC finds an empty leaf node in the current key tree. In the second scenario, the current key tree is full, and the KDC needs to create empty slots by splitting nodes. In order to maintain the balance of the key tree and accommodate as many new members as possible, the root node is generally chosen to create new slots. After the KDC associates a leaf node with the new member, all the KEKs in the internal nodes along the path from the parent node of that leaf node to the root should be changed. This key updating is necessary to ensure backward secrecy that prevents the new joining member from gaining the former group key to access the previous communication data. For the new member, the KDC encrypts the updated KEKs by the leaf node key of the new member and sends them to the new member. The KDC also sends newly-generated KEKs encrypted by the corresponding KEKs to the remaining group members who need to know the key updating.

In LKH, the rekeying messages are sent based on a user-oriented approach. For remaining group member(s), the KDC creates a rekeying message containing precisely the new keys needed by the group member(s) and encrypts the message using a KEK held by the remaining member(s). For example, in Figure 2.3, when user 3 joins the group, the KDC assigns it into leaf node 3. The leaf node key, k_3, is determined by the KDC and user 3 through the key exchange protocol such as the Internet Key Exchange (IKE) protocol [Ballardie, 1996; Harkins & Carrel, 1998;

Kaufman, 2005; Maughan & Schneider et al., 1998] or a secure communication channel. After user 3 joining the group, the internal keys (k_{34}, k_{14}, k_{18}) need to be updated to enforce backward secrecy. The KDC generates new keys $(k_{34}', k_{14}', k_{18}')$ and sends the following four rekeying messages to the remaining group members for the key updating ($\{x\}k$ means the message x is encrypted by the key k, $\rightarrow$ means unicast and $\Rightarrow$ means multicast or broadcast).

$$\text{KDC} \rightarrow \{\text{user 3}\} : \{k_{34}', k_{14}', k_{18}'\}k_3$$
$$\text{KDC} \rightarrow \{\text{user 4}\} : \{k_{34}', k_{14}', k_{18}'\}k_4$$
$$\text{KDC} \Rightarrow \{\text{user 1, 2}\} : \{k_{14}', k_{18}'\}k_{12}$$
$$\text{KDC} \Rightarrow \{\text{user 5, 6, 7, 8}\} : \{k_{18}'\}k_{58}$$

It can be observed that the number of rekeying messages sent by the KDC during the key updating process is $h+1$, where h is the height of the key tree. The number of keys encrypted by KDC during the join is:

$$1+2+...+h+h=\frac{(h+1)(h+2)}{2}-1$$

In order to reduce the overhead associated with the join operation, the Versakey framework [Waldvogel & Caronni et al., 1999] and LKH+ [Hugh Harney & Harder, 1999] have been proposed respectively. In these two approaches, instead of the KDC generating fresh keys and sending them to members, each key affected by a join operation passes through a one-way function to generate the new corresponding key. Based on this key updating scheme, every member who has the affected keys can calculate the new keys locally.

2.2.1.2 The Leave Operation

When a member leaves a group, the KDC needs to update the group key and the KEKs known to the departing member to enforce forward secrecy and prevent the leaving user from accessing future group communication. The rekeying procedure is the same as that for the join operation. For instance, for user 4 in Figure 2.3 to leave the group, keys $(k_{34}, k_{14}$ and $k_{18})$ need to be updated to the new set $(k_{34}', k_{14}', k_{18}')$. The KDC sends the following three rekeying messages to update the keys:

$$\text{KDC} \rightarrow \{\text{user 3}\} : \{k_{34}', k_{14}', k_{18}'\}k_3$$
$$\text{KDC} \Rightarrow \{\text{user 1, 2}\} : \{k_{14}', k_{18}'\}k_{12}$$
$$\text{KDC} \Rightarrow \{\text{user 5, 6, 7, 8}\} : \{k_{18}'\}k_{58}$$

The number of rekeying messages sent by the KDC during the leave process equals h, which is the height of the key tree. The number of keys encrypted by the KDC is:

$$1 + 2 + \ldots + h = \frac{h(h+1)}{2}$$

2.2.1.3 Batch and Bulk Rekeying

In a large and dynamic group, in order to enforce and maintain secure communication among the members, keys need to be updated immediately as membership changes occur. However, an immediate rekeying policy may result in frequent rekeying and overwhelm the KDC's computational capacity. In addition, the overhead associated with the rekeying communication may be seen as too costly for efficient operation. In such cases, it is possible to relax the security policy slightly to reduce the rekeying overhead. The KDC can choose to rekey the group periodically or process membership changes in batches. In batch rekeying [Li & Yang et al., 2001; Pegueroles & Rico-Novella et al., 2003; Yang & Li et al., 2001], the KDC rekeys

only after a number of group membership changes have occurred. The KDC may choose to rekey, for instance, after r membership changes occur or alternatively, after r joins or r leavings. For periodical rekeying, the KDC may rekey at a predefined interval, irrespective of membership changes.

Another way to reduce the overhead of LKH is to perform aggregation or a bulk operation to deal with several simultaneous group membership changes [Zou & Ramamurthy et al., 2005]. When several join and leave operations happen at the same time, instead of performing separate rekeying (one rekeying operation for one membership change), these changes can be processed in one bulk rekeying operation.

2.2.2 One-way Function Tree

The One-way Function Tree (OFT) has been proposed by Scherman et al. [McGrew & Sherman, 1998]. The major contribution of OFT over LKH is that it allows members to compute keys locally to reduce the communication and computation cost. OFT is, in several ways, similar to LKH. OFT applies the same tree structure as LKH to manage keys. The root key serves as a group key, and each member is associated with a unique leaf node and knows a set of KEKs from its leaf node to the root.

In OFT, the KDC does not need to send the KEKs directly to members during the key updating. Instead, the KDC distributes blinded keys that members apply to calculate locally the intermediate KEKs, including the group key. A blinded key (k_{bk}) is calculated by putting the corresponding intermediate KEK through a

one-way function *g*. For example, the blinded key of k_1 in Figure 2.4 is

$$k_{1\_bk} = g(k_1)$$

Each KEK in the intermediate node can be calculated by applying a mixing function f . Typically, f is an XOR function operating on the blinded keys of its right and left children.

$$k = f((k_{L\_bk}),(k_{R\_bk})) = f(g(k_L), g(k_R))$$

For example, the key, k_{12} , can be calculated by user 1 as

$$k_{12} = f(g(k_1), g(k_2)).$$

The key distribution rule of OFT is that each member receives the blinded keys of the siblings of nodes in the path from its associated leaf node to the root. For example, for user 1 in Figure 2.4, the KDC and user 1 decide the k_1 during the registration stage of the join operation. Then the KDC sends user 1 a set of blinded keys $(k_{2\_bk}, k_{34\_bk}, k_{58\_bk})$, which are the blinded keys of sibling nodes, (k_2, k_{34}, k_{58}), along the path from its leaf node to the root. User 1 is then able to compute the KEK, $\{k_{12}, k_{14}, k_{18}\}$, as follows:

$$k_{12} = f(g(k_1), k_{2\_bk})$$
$$k_{14} = f(g(k_{12}), k_{34\_bk})$$
$$k_{18} = f(g(k_{14}), k_{58\_bk})$$

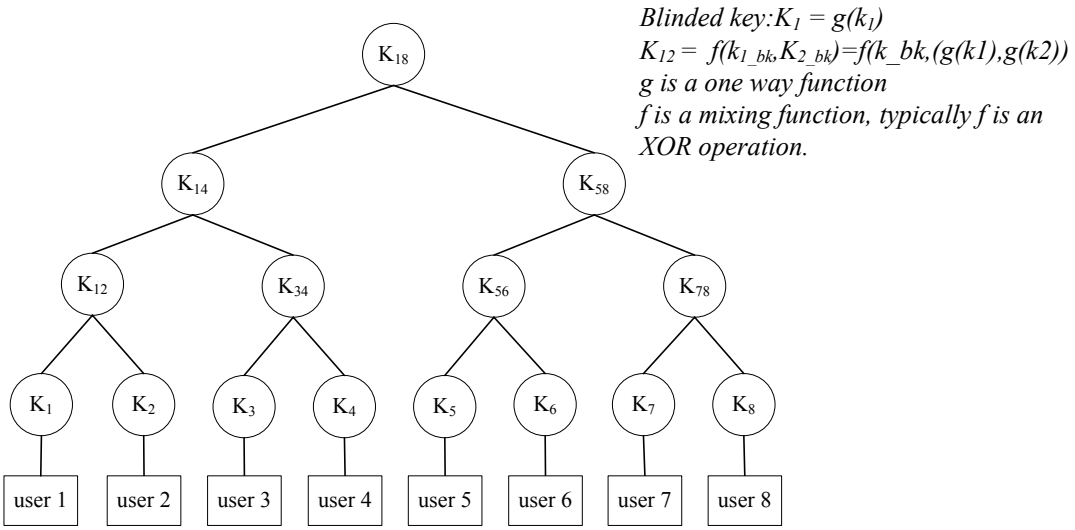


Figure 2.4 An OFT key tree

2.2.2.1 The Join Operation

When a user joins the group, the KDC and the user determine a pair-wise key associated with a leaf node. The KDC sends the new member the blinded keys it is entitled to know. Existing members also need to know the changes to the blinded keys along the path from the new member's leaf node to the root. The KDC encrypts each of these blinded keys with the corresponding sibling's KEK, and sends them to the whole group. We illustrate the rekeying process of the join operation by providing an example. For instance, user 4 in Figure 2.4 joins the group, and the KDC sends the following four rekeying messages to the whole group to update the keys:

$$\text{KDC} \rightarrow \{\text{user 4}\} : \{k_{3\_\text{bk}}, k_{12\_\text{bk}}, k_{58\_\text{bk}}\}k_4$$

$$\text{KDC} \rightarrow \{\text{user 3}\} : \{k_{4\_\text{bk}}\}k_3$$

$$\text{KDC} \Rightarrow \{\text{user 1, 2}\} : \{k_{34\_\text{bk}}\}k_{12}$$

$$\text{KDC} \Rightarrow \{\text{user 5, 6, 7, 8}\} : \{k_{14\_\text{bk}}\}k_{58}$$

When a member joins a full and balanced binary tree with *n* members (after the join operation), in OFT, the KDC needs to send the new member *h* blinded keys, where h is the height of the key tree equaling $\log_2 n$. Further, the KDC needs to send h blinded keys to the current members. Thus, the KDC needs to encrypt $2h$ blinded keys and send $h+1$ rekeying messages for a join action.

2.2.2.2 The Leave Operation

Similar to the join operation, when a member leaves the group, the KDC needs to re-compute the blinded keys known by the departing member. For example, in Figure 2.4, when user 4 leaves the group, the KDC generates a new key, k_4', for the leaf node, calculates its new blinded key, (k_{4_bk}'), and sends it to the user 3. KDC sends the new blinded keys, (k_{34_bk}', k_{14_bk}'), to the remaining members. The rekeying messages sent by the KDC during the leave operation are as follows:

$$\text{KDC} \rightarrow \{\text{user } 3\} : \{k_{4\_bk}'\}k_3$$
$$\text{KDC} \Rightarrow \{\text{user } 1, 2\} : \{k_{34\_bk}'\}k_{12}$$
$$\text{KDC} \Rightarrow \{\text{user } 5, 6, 7, 8\} : \{k_{14\_bk}'\}k_{58}$$

The KDC needs to re-calculate h blinded keys and sends h rekeying messages when a member leaves the group, where h is the height of the key tree for OFT.

2.2.2.3 Vulnerability of OFT

Although OFT can achieve communication efficiency during the membership change, it is not secure enough to prevent previous members from accessing the current group communication. OFT has been found to be vulnerable to collusion attacks [Horng, 2002; Ku & Chen, 2003]. Collusion attack refers to previous group

members cooperating to obtain the current group key to which they are not entitled. For example, in OFT, suppose a member *u* leaves the group at time t_1 and a new member *v* joins the group at time t_2, where $t_2 > t_1$. Then *u* and *v* can collude to obtain the group key to which they are not entitled during the interval $[t_1, t_2]$. Figure 2.5 shows the scenario of this possible collusion attack.

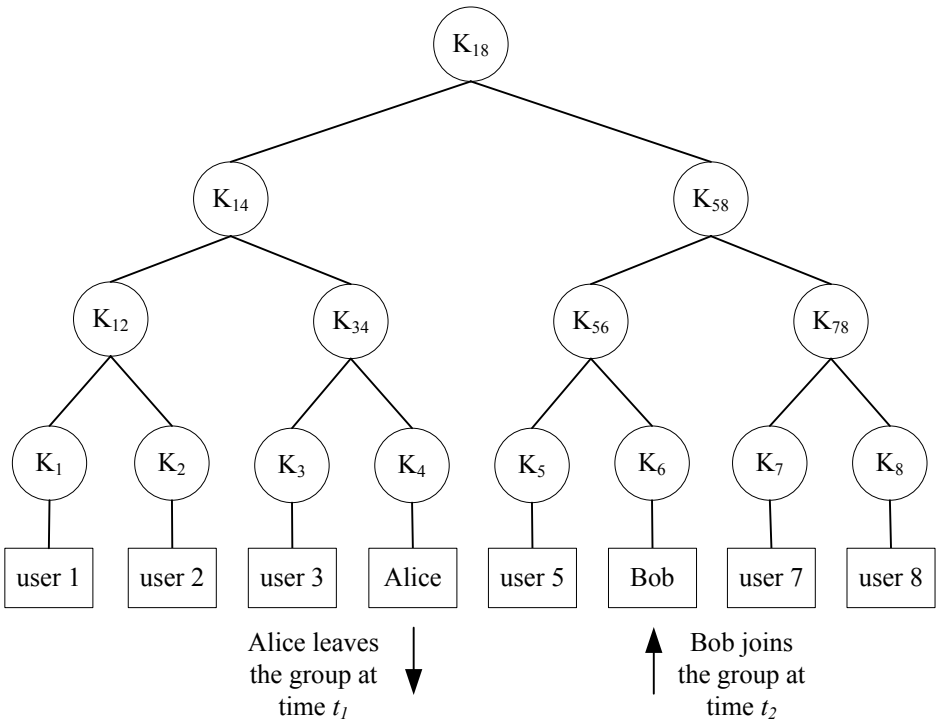


Figure 2.5 A collusion attack scenario in OFT

Initially the root key is $k = f(g(k_{14}), g(k_{58}))$. Suppose Alice, associated with leaf node 4, leaves the group at time t_1. Thus, the group key is updated to $k(t_1) = f(g(k_{14}(t_1)), g(k_{58}))$. The blinded key $g(k_{58})$ is known by Alice, and does not change at time t_1. Suppose at a later time, t_2, Bob joins the group and is associated with leaf node 6. Then the group key is updated to

$k(t_1) = f(g(k_{14}(t_1)), g(k_{58}(t_2)))$. The blinded key $g(k_{14}(t_1))$ is known to Bob. If Alice and Bob were to collude and no key updating occurred during the interval $[t_1, t_2]$, they would know the group key $k[t_1, t_2] = f(g(k_{14}(t_1)), g(k_{58}))$. As a result, OFT fails to provide key secrecy against Alice and Bob.

2.2.3 Evaluation and Summary

In the previous sections, we have introduced centralized group key management approaches (algorithms) and have investigated two prominent group key management approaches, LKH and OFT. When evaluating operational efficiency, the following three parameters can be used to measure the performance of centralized group key management approaches:

- *communication cost*: measured by the number of rekeying messages sent by the KDC during the rekeying;
- *computation cost*: assessed by the number of keys encrypted by the KDC during the rekeying procedure; and
- *key storage* cost: calculated through the number of keys stored on both the KDC and members' mobile devices.

Table 2.1 The operational costs for LKH and OFT

		LKH	OFT
Join	Communication cost	$h+1$	$h+1$
	Computation cost	$\frac{(h+1)(h+2)}{2}-1$	$2h$
Leave	Communication cost	h	h
	Computation cost	$\frac{h(h+1)}{2}$	h
Key storage cost	KDC	$2n-1$	$2n-1$
	member	$h+1$	$h+1$

h: the height of the key tree, equaling $\log_2 n$ if a binary tree is applied

n: the number of group members

From Table 2.1, we can observe that LKH and OFT can achieve the same communication efficiency at a logarithmical level due to the employment of a hierarchical tree structure. OFT has an advantage over LKH in terms of the computation cost, because OFT applies one-way function to calculate intermediate keys locally. However, this operational efficiency is at the cost of security. OFT is susceptible to collusion attacks because the KEKs in OFT are not completely independent. In contrast, LKH is able to protect itself against collusion attacks due to the complete key independence.

Centralized group key management approaches use a single trusted entity to perform and simplify key management. However, this type of approach has its weaknesses: a lack of scalability and the possibility of single-point failure. When a group increases, a single KDC may not be able to manage the larger size and a wider dispersion of members. Moreover, if the KDC is not working, the whole group is

affected and the group becomes vulnerable because the keys, which are the foundation of group security, are not being generated, regenerated and distributed.

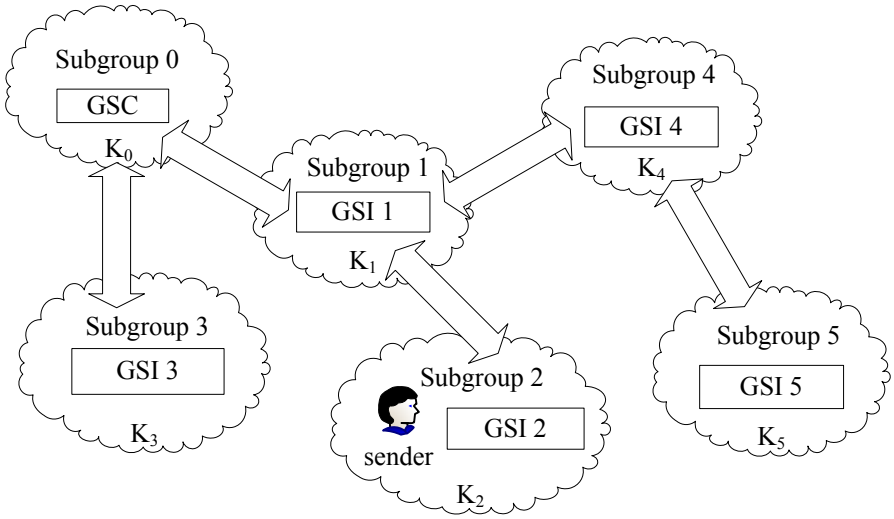
2.3 Decentralized Group Key Management Architecture

In order to minimize the problems of centralized group key management approaches such as poor scalability and single-point failure, researchers have proposed the use of decentralized group key management architecture to divide a large group into several small subgroups controlled by subgroup managers. Several group key management approaches can be classified into this type, including Core Based Tree (CBT) [Ballardie, 1996], Iolus [Mittra, 1997], Secure Transmission Backbone (STB) [Du & Ni et al., 1999], MARKS [Briscoe, 1999], Kronos [Setia & Koussih et al., 2000], Hydra [Rafaeli & Hutchison, 2002], and Inter-Domain Group Key Management Protocol (IGKMP) [Hardjono & Cain et al., 2000]. Among them, Iolus and IGKMP are the most prominent and referenced architectures. Iolus applies the a multilayered management structure while IGKMP employs a two-layered model.

2.3.1 Iolus

Iolus [Mittra, 1997] is a framework that uses a hierarchy of subgroups to perform group key management to address the scalability issue. In Iolus, a group is decomposed into a number of subgroups that form a tree structure. Each subgroup has a controller called a group security agent (GSA) or a group security intermediate

(GSI). The GSIs in the top level of the tree are managed by a group security controller (GSC). Each subgroup has its own subgroup key, A GSI builds a connection channel between its parent's subgroup and its own subgroup, so a GSI is a member of its parent's subgroup and is a member of its own subgroup as well. A GSI thus owns at least two subgroup keys. An example of Iolus is shown in Figure 2.6.



GSC: Group Security Controller
GSI: Group Security Intermediary
Ki: subgroup key

Figure 2.6 The structure of Iolus

When a sender sends a message to the group, it encrypts the message with its subgroup key and multicasts it within the subgroup. Once a GSI receives the message, it decrypts the message with the key of the current subgroup, and then encrypts the message again with its other subgroup key and forwards this message to the other subgroup. For example, a sender residing in subgroup 2, as in Figure 2.6, sends a message to the whole group. It encrypts the message with key k_2, the

encryption key of subgroup 2. When GSI 2 receives the message, it decrypts it with k_2 and encrypts it with key k_1, because GSI 2 is also a member of subgroup 1. GSI 2 then multicasts this message within subgroup 1. Once GSI 1 receives the message, it decrypts the message with k_1, re-encrypts the message with k_0 and k_4 respectively and sends the encrypted messages to subgroup 0 and subgroup 4. As a result of this process, group communication is spread within the whole domain.

The major contribution of Iolus is that Iolus minimizes the *1-affect-n phenomenon*. The 1-affect-n phenomenon refers to a single membership change affecting the entire group. In this situation, the newly-generated group key and associated supporting keys need to be distributed to all remaining members in the group to secure group communication each time a membership changes. Frequent and constant distribution of keys can have negative performance implications. In a highly-distributed group, the newly-generated keys may not reach members in time. This delay might prevent members from participating in future group communication. In military applications, communication might be interrupted during the rekeying. If the group key is updated frequently, this high-security group communication would be intermittent and the quality of service would be degraded. The 1-affect-n phenomenon is therefore a serious performance problem for group key management. In Iolus, the whole group is divided into several manageable subgroups and each group has its own subgroup key. When a member joins or leaves the subgroup, the GSI generates a new subgroup key and sends it to the remaining members within the subgroup. Key updating is restricted within the scope of the subgroup, and members outside this subgroup are not affected by the rekeying. Thus, Iolus is able to reduce

the impact of the 1-affect-n phenomenon.

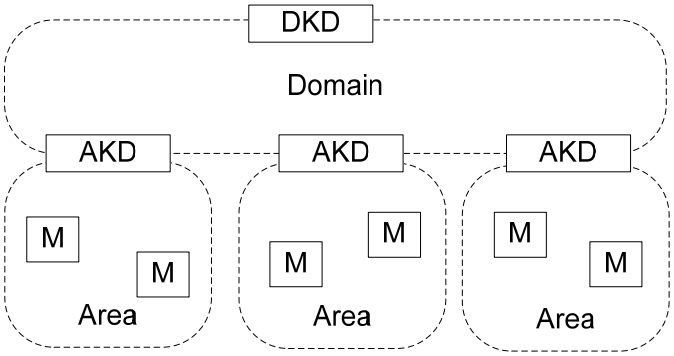
However, the Iolus structure has its weakness. Researchers have pointed out that the use of third-party entities such as GSIs may not be appropriate for many real-world applications [Hardjono & Dondeti, 2003]. A service provider sending confidential data to customers may find it unacceptable that the GSIs are able to access secret content. In order to overcome this problem and avoid GSIs gaining access to confidential data, Dondeti et al [Dondeti & Mukherjee et al., 2000; Weiler, 2001] have proposed a dual encryption protocol. This protocol establishes an additional set of keys, called key group keys, known only to group members. The sender encrypts the TEK twice, first with the key group keys and then with the subgroup key. Because the GSIs do not know the key group key, they cannot access the group communication content.

2.3.2 Intra-Domain Group Key Management Protocol (IGKMP)

The two-level Intra-Domain group key management protocol [Hardjono & Cain et al., 2000] has been proposed by Hardjono and Cain. In IGKMP, as in Iolus, the group key management domain is divided into a number of small areas. A member resides in one and only one of these small areas. Two types of Key Distributors (KD) are deployed in IGKMP.

- At the group domain level, a domain key distributor (DKD) is defined for the purpose of key management within the domain.
- At the area level, an area key distributor (AKD) is defined for key distribution within the area.

All the AKDs form a multicast group called the All-KD-Group, which is used by the DKD to transmit rekeying messages to all AKDs within the domain. When a membership change occurs, the DKD generates a new group key, and sends it to all the AKDs via the All-KD-Group. Once the AKD receives the new key, it distributes this new group key within its own area. The AKD communicates with members in its area either through a secure channel or through multicast transmission. The structure of IGKMP is shown in Figure 2.7.



DKD: Domain Key Distributor
AKD: Area Key Distributor
M: Group Member

Figure 2.7 Intra-domain group key management protocol

2.3.3 Evaluation and Summary

In this section, we have introduced the decentralized group key management architecture and have elaborated on the two foremost important decentralized architectures, Iolus and IGKMP. In order to evaluate the performance of the decentralized structure, the following parameters can be applied.

- *The number of layers*: decentralized architecture divides the whole group key management domain into several smaller areas and forms a layered structure. If the number of layers is too big, it increases the complexity of the key management structure, which affects the operational performance in terms of scalability and reliability.
- *Decentralized controller*: this refers to whether subgroup controllers are managed by a centralized controller or whether each subgroup controller can perform key management independently. If a centralized controller controls all the subgroup controllers, this raises the same issues as the centralized approaches in that the centralized controller carries the risk of single-point failure.
- *1-affect-n phenomenon*: this parameter refers to whether key updating caused by membership changes in a subgroup affects all members in the entire group.
- *Trust relationship*: this refers to whether a trust relationship is required among the group key management entities and the content providers.

Table 2.2 The comparison of Iolus and IGKM

	Iolus	IGKM
Number of layers	Multilayer (no limitation)	2
Decentralized controller	Yes, each subgroup manager is an independent KDC	No, AKDs are controlled by a centralized DKD
1-affect-n phenomenon	No, Iolus minimizes the 1-affect-n phenomenon	Yes, a single membership change affects all members
Trust relationship	Trust needs to be established if a third party is involved	Trust needs to be established if a third party is involved

Decentralized group key management architectures provide scalability and a certain level of fault-tolerance by dividing the whole group key management domain into a number of small administrative areas. This kind of approach offers a possible solution to address the scalability problem for highly-distributed and dynamic groups. However, the biggest problem of decentralized architectures is that it does not provide any key management approaches that can be efficiently applied within the subgroup. Thus, decentralized architectures need to cooperate with other group key management approaches to form a solution to address the problems of group key management.

2.4 Distributed Group Key Management Schemes

Distributed group key management schemes have been developed to provide fault-tolerance to group key management. In this type of approach, there is no KDC and group members contribute to generating the group key. Several distributed group key management schemes have been proposed, including the Group Diffie-Hellman Key Exchange (GDH) [Steiner & Tsudik et al., 1996], Octopus Protocol [Becker & Wille, 1998], STR protocol [Kim & Perrig et al., 2001], Distributed Logical Key Hierarchy [Rodeh & Birman et al., 2000] and Tree-based Group DH Key Management (TGDH) [Kim & Perrig et al., 2004a, 2004b]. In order to secure key exchange during key generation, the Diffie-Hellman (DH) key exchange protocol [Diffie & Hellman, 1976] is widely applied in these approaches. In this section, we introduce the two foremost popular and prominent distributed group key

management approaches: Group Diffie-Hellman Key Exchange (GDH) and Tree-based Group DH Key Management (TGDH).

2.4.1 Group Diffie-Hellman Key Exchange (GDH)

Steiner et al. [Steiner & Tsudik et al., 1996, 2000] have proposed three versions of Group DH Key Exchange scheme (GDH). GDH extends the two-party Diffie-Hellman key exchange protocol into a group operation. In GDH.1 and GDH.2, the overhead of computation is quite considerable due to the large number of exponentiation calculations. In order to reduce the number of exponentiation computations, GDH.3 has been proposed, in which every member only needs to perform a constant small number of exponentiation computations. GDH.3 thus offers a reduced computation cost compared to the previous two versions. There are four steps in GDH.3.

- Step 1: The first step has $n-2$ rounds and is used to collect the members' contribution from member $u_1, u_2 \cdots u_{n-2}$.
- Step 2: In the second step, member u_{n-1} broadcasts value $\alpha^{S_1 S_2 \cdots S_{n-1}}$ to all the members.
- Step 3: In this step, every member extracts its own component, s_i, from the value $\alpha^{S_1 S_2 \cdots S_{n-1}}$, and sends the result to the last member u_n.
- Step 4: Finally, member u_n raises all the received values to its secret component s_n and broadcasts these results to the members. After receiving the values from member u_n, every member can compute the group key by raising

the value to the power of its own component.

Figures 2.8 and 2.9 illustrate the GDH.3 protocol and provide an example of GDH.3 with 5 members respectively.

Stage 1 (upflow): round i, $i \in [1, n-2]$

$$u_i \rightarrow u_{i+1} : \{\alpha^{S_1 S_2 \cdots S_i}\}$$

Stage 2(broadcast): round $n-1$ (u_{n-1} sends u_n and each u_i value, $i \in [1, n-2]$)

$$u_{n-1} \rightarrow \{u_i, u_n\} : \{\alpha^{S_1 S_2 \cdots S_{n-1}}\}$$

Stage 3 (response): round n (each u_i sends a value to u_n)

$$u_i \rightarrow u_n : \{\alpha^{S_1 \cdots S_{i-1} S_{i+1} \cdots S_{n-1}}\}$$

Stage 4 (broadcast): round $n+1$ (u_n sends each u_i a value)

$$u_n \rightarrow u_i : \{\alpha^{S_1 \cdots S_{i-1} S_{i+1} \cdots S_{n-1} S_n}\}$$

Figure 2.8 GDH.3 protocol

Stage 1 (upflow):

$$u_1 \rightarrow u_2 : \{\alpha^{S_1}\}$$
$$u_2 \rightarrow u_3 : \{\alpha^{S_1 S_2}\}$$
$$u_3 \rightarrow u_4 : \{\alpha^{S_1 S_2 S_3}\}$$

Stage 2(broadcast):

$$u_4 \rightarrow \{u_1, u_2, u_3, u_5\} : \{\alpha^{S_1 S_2 S_3 S_4}\}$$

Stage 3 (response):

$$u_1 \rightarrow u_5 : \{\alpha^{S_2 S_3 S_4}\}$$
$$u_2 \rightarrow u_5 : \{\alpha^{S_1 S_3 S_4}\}$$
$$u_3 \rightarrow u_5 : \{\alpha^{S_1 S_2 S_4}\}$$
$$u_4 \rightarrow u_5 : \{\alpha^{S_1 S_2 S_3}\}$$

Stage 4 (broadcast):

$$u_5 \Rightarrow \{u_1, u_2, u_3, u_4\} : \{\alpha^{S_2 S_3 S_4 S_5}, \alpha^{S_1 S_3 S_4 S_5}, \alpha^{S_1 S_2 S_4 S_5}, \alpha^{S_1 S_2 S_3 S_5}\}$$

Figure 2.9 An example of GDH.3 with 5 members

From Figures 2.8 and 2.9, it can be observed that the communication and computation costs for a group with *n* member can be described as follows.

- The number of rounds (which refers to the number of iterations among the members) is $n+1$.
- Each member (except the last two members, u_{n-1} and u_n) performs exponentiation computations three times. Member u_{n-1} performs two exponentiations computations and the last member, u_n, performs *n* times exponentiations computations.
- The total number of messages sent during the key generation is $2n-1$.
- Each member (expect the last member) sends two keying messages during the

key generation to make its own contribution. The last member broadcasts one message to the group.

- Each member (except the last two members) receives three keying messages. Member u_{n-1} receives two messages and the last member, u_n, receives n keying messages.

A problem of GDH.3 is that the last member in the group is a special user whose performance determines the success or failure of the group key generation. In GDH.3, the last group member receives n messages and needs to perform n exponentiations computations. This requires the member to have plenty of storage space and strong computational power. However, not every member has such power, especially in wireless networks. Furthermore, the time taken for the key generation significantly increases with the growth of the group size. This increase causes a slow response to membership changes.

2.4.2 Tree Based DH Key Management (TGDH)

Kim et al. [Kim & Perrig et al., 2004b] have proposed a Tree Based Group Diffie-Hellman key management (TGDH) to extend the two-party DH protocol to a hierarchical structure to secure multi-party communication. In TGDH, all members maintain an identical virtual binary tree that may or may not be balanced. Each member is associated with a leaf node in the key tree. Instead of applying a one-way function to generate the keys in the upper level, members use Diffie-Hellman protocol to generate the keys along the path from its leaf node to the root. The key of

each internal node is generated from its two children ($k = \alpha^{k_L k_R} \bmod p$). Each member chooses its own secret key (k_s), calculates a public shared blinded key $k_{bk} = \alpha^{k_s}$ and broadcasts this blinded key to the group. An example is provided to illustrate TGDH further (Figure 2.10).

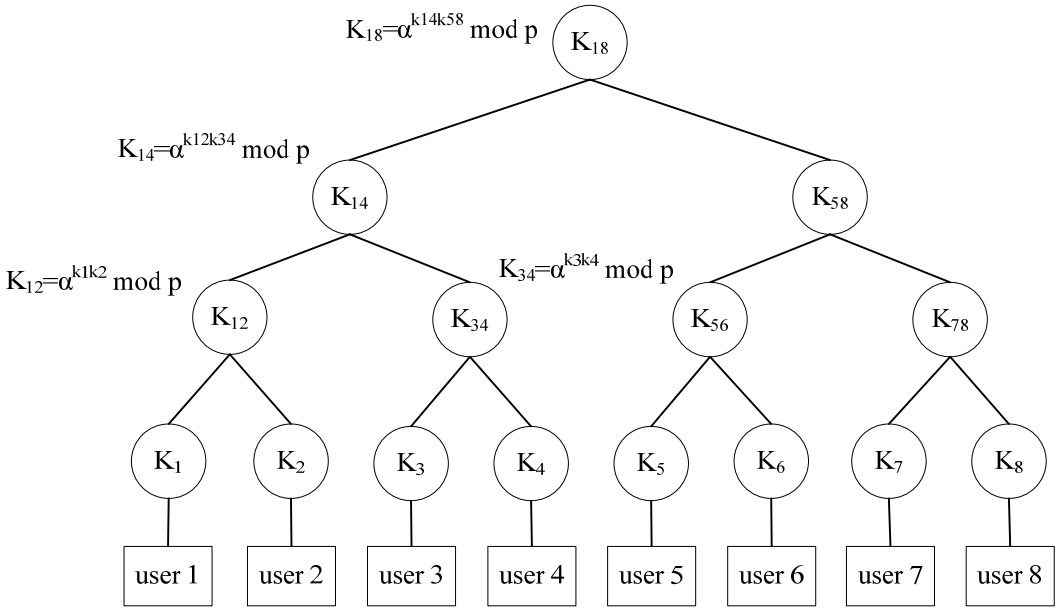


Figure 2.10 TGDH protocol

In Figure 2.10, user u_2 and u_1 agree to use the key k_{12} by applying DH protocol ($k_{12} = \alpha^{k_1 k_2} \bmod p$). The same process is applied to keys $k_{34} = \alpha^{k_3 k_4} \bmod p$, $k_{56} = \alpha^{k_5 k_6} \bmod p$ and $k_{78} = \alpha^{k_7 k_8} \bmod p$. This algorithm results in $k_{14} = \alpha^{k_{12} k_{34}} \bmod p$ and $k_{58} = \alpha^{k_{56} k_{78}} \bmod p$. Finally, the group key can be calculated as $k_{18} = \alpha^{k_{14} k_{58}} \bmod p$.

In TGDH, the rekeying for the join and leave operations are both performed in the same way. A sponsor is responsible for the rekeying of the join and leave operations. A sponsor is a member with the following attributes:

- a sponsor of a subtree is a member hosted on the rightmost leaf in the subtree,
- a sponsor of a leaf node (for example, leaf node v) is the member hosted on the rightmost leaf node (other than itself) of the lowest subtree to which v belongs.

When a user, *u*, wishes to join the group, *u* broadcasts its public shared blinded key to the entire group. Next, all current members determine the insertion location for *u* on the tree and the sponsor of *u*. Each member updates the maintained key tree by adding a new leaf node for u and a new internal node, and removes all secret keys and blinded keys from the sponsor's leaf node to the root node. Then, the sponsor generates its new secret key, computes all blinded keys from its leaf node to the root, and broadcasts all these new blinded keys to the whole group. Every member is able to calculate the group key after receiving these new blinded keys. We provide an example to further illustrate the key updating procedure of TGDH.

When user 1 in Figure 2.10 joins the group, user 1 generates its new secret key k_1, and calculates the shared blinded key $k_{1\_bk} = \alpha^{k_1}$. User 1 broadcasts this blinded key to the entire group. User 2 is chosen as the sponsor according to the sponsor policy. After user 2 receives the blinded key from user 1, it generates its new secret key k_2 and calculates the key k_{12} and its corresponding shared blinded key k_{12_bk}.

$$k_{12} = \alpha^{k_1 k_2} \bmod p$$
$$k_{12\_bk} = \alpha^{k_{12}} = \alpha^{\alpha^{k_1 k_2}}$$

As user 2 knows the blinded keys of key k_{34} and k_{58} from the previous key updating, user 2 starts to compute the rest of the intermediate keys and its corresponding

blinded keys from its leaf node along the path to the root.

$$k_{14} = \alpha^{k_{12}k_{34}} \text{ mod p}$$
$$k_{14\_bk} = \alpha^{k_{14}}$$
$$k_{18} = \alpha^{k_{14}k_{58}} \text{ mod p}$$

User 2 broadcasts these blinded keys $(k_{2\_bk}, k_{12\_bk}, k_{14\_bk})$ to the whole group.

$$\text{user 2} \Rightarrow \text{ user 1, 2,....,8: } \{k_{2\_bk}, k_{12\_bk}, k_{14\_bk}\}$$

After receiving these blinded keys, current group members can calculate the intermediate keys locally.

For user 1:

$$k_{12} = \alpha^{k_1 k_2} \text{ mod p} = (k_{1\_bk})^{k_2} \text{ mod p}$$
$$k_{14} = (k_{34\_bk})^{k_{12}} \text{ mod p}$$
$$k_{18} = (k_{58\_bk})^{k_{14}} \text{ mod p}$$

For users 3 and 4:

$$k_{14} = (k_{12\_bk})^{k_{34}} \text{ mod p}$$
$$k_{18} = (k_{58\_bk})^{k_{14}} \text{ mod p}$$

For users 5, 6, 7 and 8:

$$k_{18} = (k_{14\_bk})^{k_{58}} \text{ mod p}$$

From the above TGDH scheme and example, we can observe that TGDH is operational efficient in communication and computation, because only one round is required to calculate the group key. The sponsor only needs to send one keying message. The keying message contains $\log_2 n$ blinded keys (n is the number of users in the group). The members in the group perform, at most, $\log_2 n$

exponentiations computation to reach the group key. However, TGDH relies on a special member, a sponsor. If the sponsor fails, the whole key updating procedure stops. Moreover, in TGDH, each member needs to maintain an identical virtual binary tree. Substantial, storage space is required to maintain such a tree structure for a large group. This high storage requirement may not be available to everyone in wireless networks.

2.4.3 Evaluation and Summary

In the previous sections, we have discussed the distributed group key management schemes and have investigated two well-known approaches, GDH and TGDH. We can apply the following parameters to evaluate the distributed group key management schemes:

- *the number of rounds*: the number of iterations among members;
- the total number of messages sent during the rekeying;
- the number of messages sent and received by each member;
- the number of exponentiations computations performed by each member; and
- *special group member*: the rekeying depends on one or more key member(s) to be performed.

Table 2.3 The comparison of GDH and TDGH

	GDH.3	TDGH
Number of rounds	$n+1$	1
Total number of messages	$2n-1$	1
Number of messages sent per member	member (except last one): 2 last member: 1	new joining member: 1 sponsor:1
Number of messages received per member	Member(except last one): 3 Last second member: 2 Last member: $n-1$	1
Number of exponentiation computations performed per member	Member (except last two): 4 Last second member: 2 Last member: n	at most $\log_2 n$
Special member(s)	Last member in the group	sponsor

n: the number of members in the group

From Table 2.3, we can conclude that TDGH has an advantage over GDH.3 in communication, because the total number of messages sent in TDGH is only one. However, TGDH requires all members to maintain an identical virtual binary tree. It is difficult to achieve this synchronization in the real world. Moreover, both of these approaches depend on a specific member to perform rekeying. The member could become a performance bottleneck.

2.5 Network-Independent Group Key Management in Wireless Networks

In order to provide universality, network-independent group key management approaches do not take the underlying network architecture into consideration. However, when they are applied to the wireless environment, due to the limitations of wireless networks and mobile devices, these approaches encounter problems and challenges that prevent them from operating efficiently. In this section, we analyze the performance of these network-independent group key management schemes in wireless networks. We begin by investigating centralized key management approaches, followed by decentralized architecture and finally distributed group key management schemes.

2.5.1 Centralized Group Key Management in Wireless Networks

Centralized group key management schemes, such as LKH and OFT, apply a single KDC and a hierarchical key structure to facilitate key management during key distribution and updating. When it is employed in the wireless environment, the centralized approach encounters three major problems: (i) lack of scalability, (ii) communication and computation inefficiency and (iii) inability to handle multiple-membership changes.

In a large and highly dynamic wireless group application, frequent rekeying may overwhelm the capacity of a single KDC and cause the failure of key management operations. This failure would jeopardize the security of group

application. Moreover, as the number of group users increases, members need to process a great number of rekeying messages. The frequent keying that would result from a large group with high-dynamic membership changes could overwhelm the capacity of lightweight mobile devices. The inability to cope with increasing group size – the lack of scalability - is the first problem encountered by centralized schemes in wireless networks.

When a centralized key management approach is applied in the wireless environment, due to not knowing the location of users, rekeying messages have to be multicasted within the entire wireless domain to be delivered to the affected members. For example, in the cellular wireless network, rekeying messages have to be multicasted by the base station within each cell to locate the targeted receivers. However, in LKH and OFT, each rekeying message (as discussed in section 2.2.2) is only useful to a portion of the members. Therefore, when rekeying messages are multicasted within the whole wireless domain, many rekeying messages are discarded by group members because the rekeying messages are irrelevant. This rekeying process thus wastes scarce wireless communication bandwidth. Furthermore, each member needs to process each received rekeying messages in order to find the single message useful to it. Again, this process wastes communication and computation resources. The centralized group key management scheme is thus unable to perform efficiently in the wireless environment.

Another performance issue for the centralized group key management approaches is multiple-membership changes, whereby a member changes memberships between several group applications from the same service provider. In

traditional centralized schemes, the key management structure is linked to a specific group application and a separate key tree is established for each group application. In order to participate in several applications, a member has to register itself with several key trees. A number of keys are then required to be stored and managed for the member. Furthermore, several key trees and all the members residing in these key trees are affected when multiple-membership changes occur. Many rekeying messages need to be broadcasted within the wireless network during the rekeying. This kind of approach is inefficient for resource-restricted wireless networks as it increases the storage, communication and computation burden for both the KDC and members. A wireless group key management approach therefore needs to tackle the problem of multiple-membership changes efficiently and intelligently.

2.5.2 Decentralized Architecture in Wireless Networks

Decentralized architecture provides a useful approach to tackle the scalability issue for group key management in a large area by dividing the whole group into several small administrative subgroups. These schemes are therefore suitable to support group key management for large-scale wireless networks such as cellular wireless network, WiMax [Ahson & Ilyas, 2007; Radha Krishna Rao & Radhamani, 2008] and the future 4G system [Glisic, 2006]. Furthermore, decentralized architecture is the only solution to address the 1-affect-n phenomenon that also needs to be considered when dealing with wireless networks. However, decentralized architecture only proposes a framework for large-scale group key management; it does not provide an approach for efficiently distributing keying materials to group

members in subgroups. Therefore, decentralized architecture needs to cooperate with other group key management approaches to provide an integrated solution for group key management in wireless networks. Furthermore, in wireless networks, third-party entities are commonly involved in decentralized architectures. The wireless network operator and secure group application providers are generally different entities. Establishing a trust relationship between them is a critical security concern that also needs to be addressed by wireless group key management approaches.

2.5.3 Distributed Group Key Management in Wireless Networks

Distributed group key management schemes abolish group controllers in order to avoid the possibility of single-point error that is associated with centralized group key management approaches. Instead, distributed group key management approaches provide fault-tolerance by using member contributions to generate group key. Distributed group key management approaches allow all group members to compute the same group key independently and locally. Any single member failure can be ignored, because it does not prevent the other members from reaching a shared group key. However, this fault-tolerance feature sacrifices operational efficiency at the cost of communication and computation. While the Diffie-Hellman key exchange protocol is widely applied in distributed schemes in order to derive a group key, its expensive exponentiation computation may not be affordable to mobile devices. Furthermore, it takes a long time for all group members to reach a group key. Along with the growth in the group size, the convergence time of group key generation

increases. Distributed group key management approaches are therefore not suitable for large and highly dynamic secure group applications.

2.5.4 Summary

The capacity limitations of wireless networks and mobile devices have performance implications for network-independent group key management schemes in the wireless environment. The most serious problem is that of operational efficiency. To be considered effective, a wireless group key management scheme needs to minimize the communication, computation and key storage overhead for both the KDC and members. In addition, other problems and challenges such as the need for scalability, the 1-affect-n phenomenon and the trust relationship issue also need to be addressed by the wireless group key management approach. The ability of centralized, decentralized and distributed group key management schemes in wireless networks to cope with these issues is summarized in Table 2.4.

Table 2.4 The comparison of the three group key management types

	Advantages	Disadvantages
Centralized group key management	• Simple management model • Logarithmical level of communication computation and key storage cost • Easy implementation	• Scalability issue makes it unsuitable for large and highly-dynamic wireless group applications • Communication inefficiency during rekeying • Inefficiency when dealing with multiple-memberships
Decentralized group key management architecture	• Provides a framework to address group key management for large-scale wireless networks • Able to address the 1-affect-n phenomenon	• Cannot work alone. It needs to cooperate with other group key management approaches to form a comprehensive solution • Third party trust relationship is required
Distributed group key management	• Provides fault-tolerance to avoid single-point error	• Expensive operation cost • Not affordable to all mobile devices and not suitable for wireless networks

2.6 Network-Dependent Wireless Group Key Management Schemes

In order to perform group key management in wireless networks efficiently, wireless group key management approaches need to consider the underlying wireless network environment to facilitate key management. However, the variety of wireless networks poses a new challenge to group key management. Each type of wireless

networks has unique features. For example, the cellular wireless network has a distributed topology and powerful base stations, whereas the wireless sensor network, compared to other wireless environments, has limited communication, computation and storage capacity [Callaway, 2004]. A wireless group key management system cannot be compatible with all wireless networks. A wireless group key management approach needs thus to be tailored to a certain type of wireless network. Researchers have consequently focused attention on the prominent cellular wireless network when developing group key management schemes. In the following section, we introduce one such group key management scheme.

2.6.1 Topology-Matching Key Management (TMKM)

Sun et al [Sun & Trappe et al., 2003, 2004] have proposed the group key management scheme Topology Matching Key Management (TMKM) to extend the centralized group key management approach to the wireless environment. The approach adapts the traditional key tree (that is the LKH key tree) to the cellular wireless network and matches the key tree to a three-level topological structure.

In TMKM, key management is based on a cellular wireless network model [Brown & Singh, 1998], which consists of mobile users, base stations (BS) and a supervisor host (SH), as shown in Figure 2.11. The SH is also a part of the wired network and handles most of the routing and protocol detail for mobile users, controls base stations and manages the keying materials including the group key and the supporting keys to protect group communication.

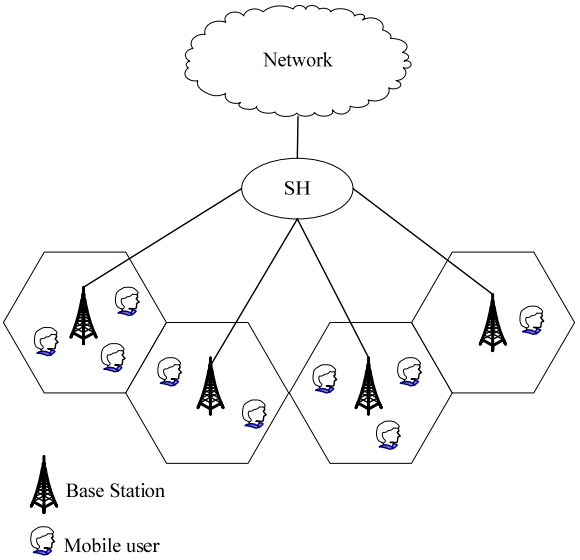


Figure 2.11 A cellular wireless network model

TMKM applies a hierarchical structure to manage the keys. This key management tree is designed in three steps from the bottom to the top (Figure 2.12).

(i) TMKM designs a subtree for the users under each BS. This subtree is called user-subtree.

(ii) TMKM designs a subtree that governs the key hierarchy between BSs and SHs. This subtree is called BS-subtree.

(iii) TMKM designs a subtree that governs the key hierarchy between SHs and KDC. This subtree is called SH-subtree.

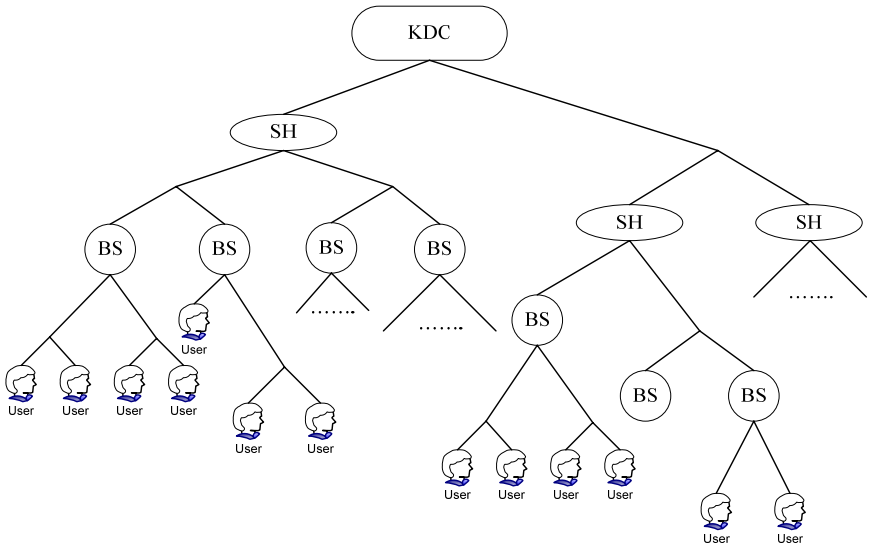


Figure 2.12 A TMKM three-level structure

Each BS is responsible for performing key management within its scope and multicasts the keying material to users in its cell. It is assumed that SH and BSs know whether a rekeying message is useful to its users. There are three steps to distribute the rekeying message.

- Step 1: The KDC multicasts the rekeying message to all SHs via wired networks.
- Step 2: The SH multicasts the rekeying message to all BSs through wired networks when it is useful to its users.
- Step 3: Each BS broadcasts the rekeying messages to its subgroup, when it finds it is useful to the users.

When a user joins or leaves the group, the rekeying operation is the same as the one described for LKH. Instead of broadcasting the rekeying messages to the entire group, the rekeying messages are only delivered to the user who needs them, with the assistance of SHs and BSs.

TMKM introduces an (α, L, x)-logic tree for the user-subtree (Figure 2.13).

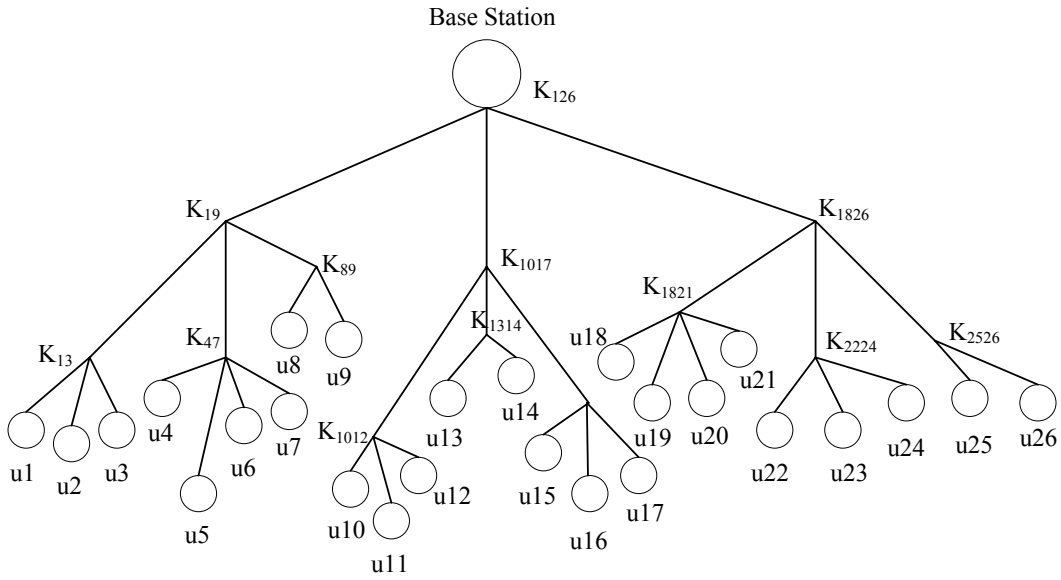


Figure 2.13 A ALX tree

The (α, L, x)-logic tree is defined as follows: (i) it has $L+1$ levels; (ii) the upper L levels form a subtree with degree α and this L levels tree is fixed during the whole group communication; and (iii) members are associated with the nodes at $(L+1)$th level that has no fixed degree and changes when membership change occurs. A vector $x = \{x_1, \cdots, x_i, \cdots, x_{\alpha L}\}$ is applied to describe the members in this level, where x_i is the number of members attached to the *i*th node at level *L*. For example, as shown in Figure 2.13, the vector *x* is $\{3,4,2,3,2,3,4,3,2\}$. It is observed that the ALX tree is a derivative of α-tree. Therefore, we can calculate the operational cost of the join and leave operations based on the α-tree. For example, when user 9 in Figure 2.13 joins the group, the BS needs to generate new keys (k_{126}', k_{19}' and k_{89}') and sends four rekeying messages to update keys in the (α, L, x)-logic tree.

$$BS \Rightarrow \{u10 \cdots u26\} : \{k_{126}'\} k_{126}$$
$$BS \Rightarrow \{u1 \cdots u7\} : \{k_{126}', k_{19}'\} k_{19}$$
$$BS \Rightarrow \{u8\} : \{k_{126}', k_{19}', k_{89}'\} k_8$$
$$BS \Rightarrow \{u9\} : \{k_{126}', k_{19}', k_{89}'\} k_9$$

We can observe that the communication and computation cost of the join process are $L+1$ and $1+2+\cdots+L+L=\frac{(L+1)(L+2)}{2}-1$ respectively, where L is the number of levels in ALX tree.

When a member leaves the group (for instance, if user 26 leaves the group), the BS needs to generate a new set of intermediate keys (k_{126}', k_{1826}' and k_{2526}') and BS sends five rekeying messages to update keys for the affected members.

$$BS \Rightarrow \{u1 \cdots u9\} : \{k_{126}'\} k_{19}$$
$$BS \Rightarrow \{u10 \cdots u17\} : \{k_{126}'\} k_{1017}$$
$$BS \Rightarrow \{u18 \cdots u21\} : \{k_{126}', k_{1826}'\} k_{1821}$$
$$BS \Rightarrow \{u22 \cdots u24\} : \{k_{126}', k_{1826}'\} k_{2224}$$
$$BS \Rightarrow \{u25\} : \{k_{126}', k_{1826}', k_{2526}'\} k_{25}$$

Therefore, it can be observed that the communication cost of the leave operation is $(\alpha-1)(L-1)+(x_i-1)$, where α is the degree of the ALX tree, L is the number of levels of ALX key tree and x_i is the number of members in the ith node.

In terms of the key storage cost, the KDC needs to keep $\frac{\alpha n-1}{\alpha-1}+\sum_i^n x_i$ keys, where $\frac{\alpha n-1}{\alpha-1}$ is the number of keys for the L-level and α-degree key tree, and $\sum_i^n x_i$ is the total number of members in the (L+1)th level.

The operational costs of (α, L, x)-logic tree is tabulated in Table 2.5.

Table 2.5 The operational costs of ALX tree

	Join	Leaving
Communication cost	$L+1$	$(\alpha-1)(L-1)+(x_i-1)$
Computation cost	$\frac{(L+1)(L+2)}{2}-1$	$(\alpha-1)(L-1)+(x_i-1)L$
Key storage cost	KDC:	$\frac{\alpha n-1}{\alpha-1}+\sum_i^n x_i$
	member:	$L+1$

α : the degree of (α, L, x) -tree

L : the number of levels of (α, L, x) -tree

x_i : the number of members in the node i

The contribution of TMKM to group key management is that TMKM matches the key tree structure to the cellular wireless network topology to reduce the communication overhead during the rekeying procedure. In TMKM, rekeying messages are only sent to the wireless cell in which the affected members reside. This approach improves the performance of group key management in wireless networks compared with traditional centralized group key management approaches that needs to broadcast rekeying messages within the whole wireless network.

However, TMKM still has the following limitations that need to be addressed.

- TMKM suffers from the 1-affect-n phenomenon. As only one group key is applied to the whole group, when membership changes, rekeying affects the entire group.
- Although TMKM improves communication efficiency during the rekeying, communication and computation are still inefficient for the BSs and user

subtrees during the rekeying. Each BS needs to process all the rekeying messages sent to it to determine whether the messages are useful to its users. In the user subtree, the BS needs to multicast all rekeying messages to the members within the cell to achieve key updating, while each group member still receives all the rekeying messages and needs to process all of them to find its own message. During these processes, system resources are wasted.

- TMKM has a trust relationship issue. Generally, BSs and SHs are provided by a third party. However, the content provider may not be comfortable with BSs and SHs having access to confidential keying materials.

2.7 Summary

In order to enforce access control over group communication, a shared group key is applied to encrypt group data. The security of group communication relies on the safety of this group key. Group key management is necessary to control key generation, distribution and updating. In this chapter, we have investigated current group key management approaches. These approaches can be categorized into two classes: network-independent and network-dependent. Network-independent approaches can be further classified into three types: centralized, decentralized and distributed group key management schemes. Each of these has its own advantages and problems when it is applied in the wireless environment. Centralized group key management approaches can achieve a logarithmical level overhead in communication and computation, but they still have an operational efficiency issue

when applied in the wireless environment. In addition to this, scalability and multiple-membership problems trouble the centralized key management approaches, and prevent them from being deployed in a large-scale wireless environment. Decentralized architecture provides a framework to deal with group key management in a large-scale wireless network; however, it needs to cooperate with other group key management approaches to achieve efficient operation. A distributed group key management scheme offers fault-tolerance, but at the cost of communication and computation outlays. Such outlays are beyond the reach of most lightweight capacity mobile devices.

In conclusion, group key management in the wireless environment faces several serious problems and challenges. Of these, operational efficiency is the most important issue. Figure 2.14 summarizes these problems and challenges of group key management in wireless networks.

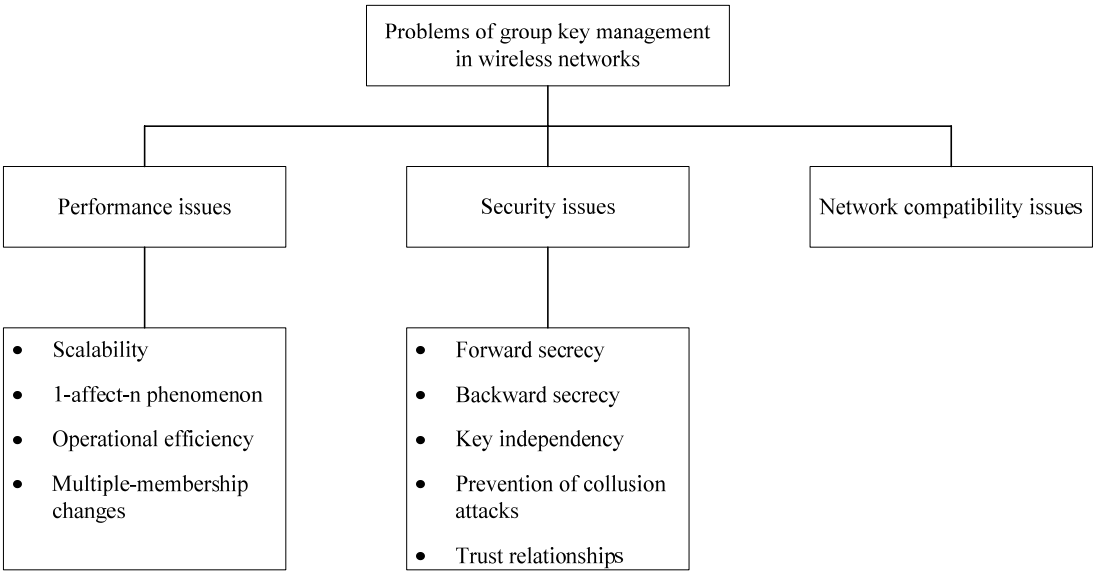


Figure 2.14 Problem areas in wireless group key management

Existing group key management approaches are technically incapable of being applied successfully to wireless networks. This situation motivates us to conduct this research to propose a new group key management solution for the wireless environment. In the next chapter, we propose a new model for wireless group key management systems to overcome the identified problems.

Chapter 3

A Group Key Management System Model for Wireless Networks

3.1 Introduction

In the previous chapter, we have demonstrated that the problems of group key management in wireless networks mainly pertain to three perspectives: performance, security and network compatibility. In order to systematically identify and address these problems, a formal model is required. Although the IETF model [Baugher & Canetti et al., 2005; Hardjono & Tsudik, 1999] (discussed in section 2.1) is considered, this model is not suitable for wireless networks because it does not take into account the role and function of the underlying network infrastructure in group key management systems. Due to the resource limitations of wireless networks and mobile devices, a group key management system needs to cooperate closely with the

underlying wireless network to utilize the network's capacity to efficiently perform group key management. Therefore, in this chapter, we propose a new group key management system model dedicated to wireless networks. The major contribution of this model is to take into account the role and function of the wireless environment in a group key management system. The purpose of this model is to identify the key components in wireless group key management system and provide a guideline for the design and evaluation of wireless group key management systems.

Of the existing wireless networks - cellular wireless networks, wireless LANs and satellite networks - the cellular wireless network is the most important and dominant network topology [Pahlavan & Krishnamurthy, 2001]. It is also the most widely deployed wireless communication system in the world. In this thesis, due to the importance of the cellular wireless network and the network compatibility of group key management systems, we focus our research on the cellular wireless network. Based on the proposed wireless group key management model, we propose a group key management architecture for the cellular wireless network in the latter part of this chapter. The purpose of this architecture is to match the group key management structure with the cellular wireless network topology and fully utilize the capacity of the cellular network to facilitate group key management.

This remainder of this chapter is organized as follows. Section 3.2 investigates the proposed model to identify the important components in wireless group key management systems. Section 3.3 delineates the proposed group key management architecture for the cellular wireless network and also analyzes the proposed architecture to demonstrate its capacity to tackle the group key management

problems. Finally, section 3.4 summarizes this chapter.

3.2 A Formal Model for Wireless Group Key Management Systems

3.2.1 The Proposed Model

Due to the inadequacies of the IETF group key management model when applied to wireless networks, we propose a new model, shown in Figure 3.1, to identify the key components and issues in wireless group key management systems. In this model, we take the underlying wireless network infrastructure into consideration to identify its role and function in a wireless group key management system. We define three major components in the wireless group key management system in this model: (i) the wireless network environment, (ii) the wireless group key management network architecture, and (iii) the group key management operation unit.

- Wireless network environment

The role and function of the wireless network in a group key management system are critical because they provide not only the communication function for the group key management system but also the network compatible features for the other components to facilitate key management. The underlying wireless network needs to be considered before addressing the network-compatible issues in wireless group key management systems.

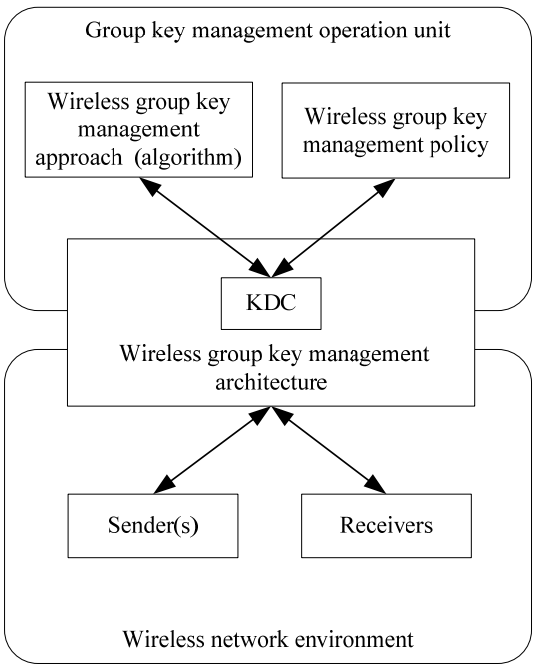


Figure 3.1 The formal model for wireless group key management systems

In the previous chapter, we have investigated several network-independent group key management approaches. When these are applied in wireless networks, the rekeying messages have to be multicasted in the whole wireless network domain to find the targeted receivers. This is inefficient. To avoid this problem, a group key management system needs to cooperate closely with the underlying wireless network to efficiently perform key management to reduce such rekeying overhead. Each type of wireless networks (for example, the cellular wireless network and the wireless sensor network) has its own unique properties. The cellular wireless network is a large-scale wireless network that applies a distributed network topology. Each cell is managed by a powerful base station. On the other hand, the wireless sensor network has limited capacity for communication, computation and storage, no centralized controller in the sensor network and uses an ad hoc network topology [Callaway,

2004]. The operation model of a sensor network is independent and cooperative. These network-specific features of each wireless network affect the design and operation of group key management systems. Due to the varying characteristics of wireless networks, no single group key management system can fit all types of wireless networks. Therefore, a wireless group key management system needs to be developed based on the underlying wireless environment to address the network-compatible issue.

In summary, in order to tackle the network-compatible issues and perform group key management efficiently in a specific wireless environment, it is necessary to understand the underlying wireless environment. Once the network-specific features are identified, these can be utilized in the development of group key management systems.

- Wireless group key management architecture

Wireless group key management architecture is defined as the topology and configurations of entities that are involved in group key management in the wireless network [Hardjono & Dondeti, 2003]. The purpose of this architecture is to establish a key management architecture based on the underlying wireless network to facilitate group key management. It aims to tackle the problems of scalability and the 1-affect-n phenomenon at the infrastructure level. Several entities – key distribution centers (KDCs), access points and base stations - are involved in wireless group key management. Wireless group key management architecture provides an infrastructure based on the underlying wireless environment to arrange and manage these key management entities, especially KDCs. locating and configuring these entities to

facilitate key management is a critical issue. The basic principle is to locate the entities as close to the users as possible and accommodate as many group members as possible. Therefore, wireless group key management architecture needs to be established to utilize the capacity of the underlying wireless network to fulfill this principle. Moreover, this architecture also provides a platform to house other group key management components such as group key management operation unit.

- Group key management operation unit

The group key management operation unit is the most important module in wireless group key management systems because this entity performs the key management tasks of key generation, key distribution and key updating. The group key management operation unit can be divided into two main components: the wireless group key management approach (algorithm) and the wireless group key management policy.

  - The wireless group key management approach (algorithm) refers to the method of arranging, managing and updating keying materials including the group key and all supporting keys that are used to manage the group key. The group key management approach is the central core of a group key management system, as the key management tasks (key generation, key distribution and key updating) need to be efficiently performed according to a particular key management algorithm. In the wireless environment, operational efficiency is the first requirement for group key management systems due to the limited resources. The operational efficiency of the group key management approach determines the performance of a wireless group

key management system. Therefore, the group key management approach is crucial to tackling the problems of operational efficiency in the wireless group key management system.

- The wireless group key management policy is defined as a set of rules that are required to govern the behavior of engaging entities during group initialization, key distribution, membership changes and so on [Hardjono & Harney, 2002; Hugh Harney & Colgrove et al., 2001]. Because this thesis focuses on the group key management approach, the issue of group key management policy is not discussed in depth.

In addition to the three essential components of wireless group key management systems, data flow in wireless group key management system is also identified in this model, shown as thick double-arrow lines in Figure 3.1. Sender(s) and receivers connect themselves to the wireless network and send the key management request to the KDC via the wireless network. When the KDC receives these requests, it invokes the corresponding operation unit to respond to the requests. The replies are sent back from the KDC to the group members via the wireless network.

3.2.2 System Evaluation Criteria

In order to design efficient, secure and practical wireless group key management systems, system designers need a set of assessment parameters for analysis and evaluation. Based on the proposed model and the problems of wireless group key management, we propose that wireless group key management systems can be

analyzed and evaluated from three aspects: performance, security and network compatibility, as shown in Figure 3.2.

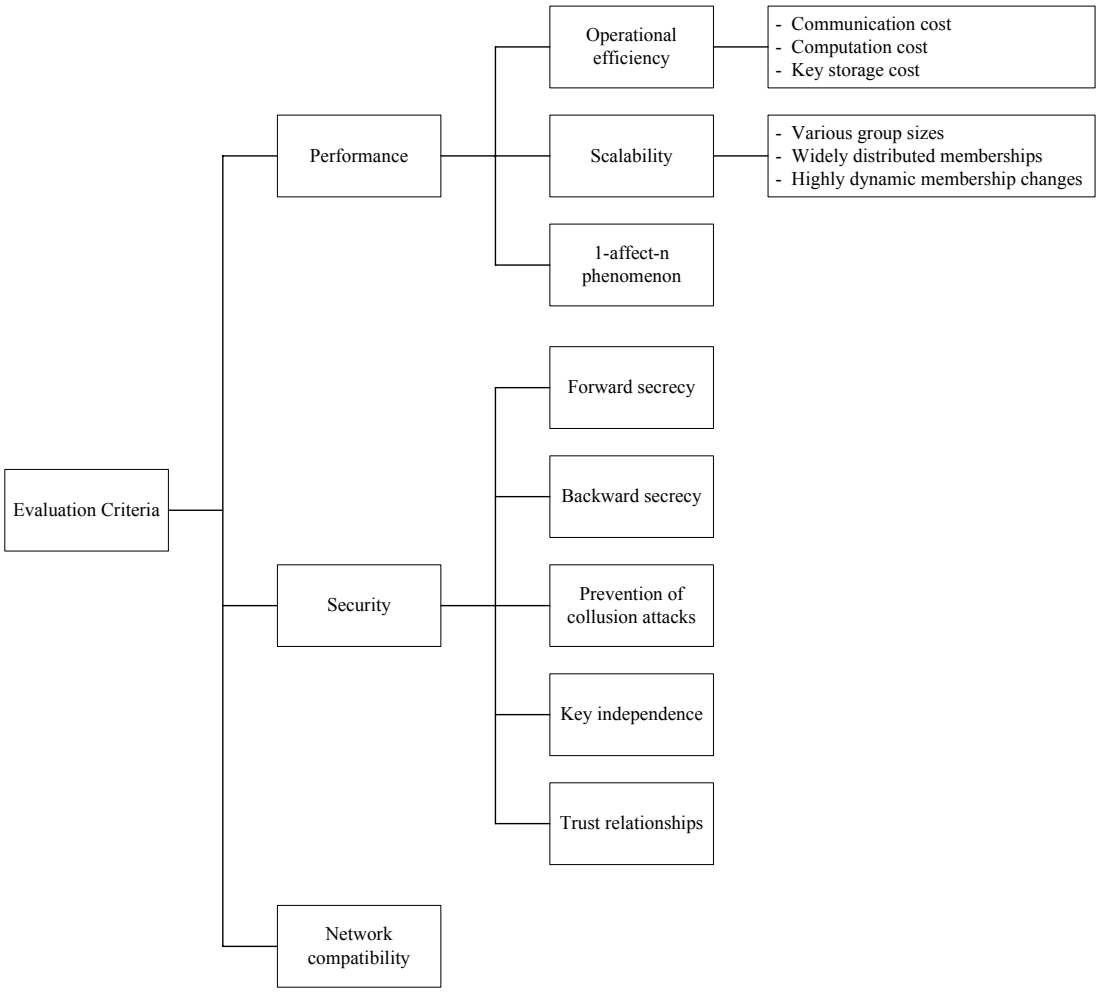


Figure 3.2 Evaluation criteria of wireless group key management

Performance criteria are used to measure the operational efficiency and practicability of wireless group key management systems. It can be further classified into the following three perspectives: operational efficiency, scalability and the 1-affect-n phenomenon [Canetti & Garay et al., 1999; Challal & Seba, 2005; Mittra, 1997].

- *Operational efficiency*

Operational efficiency refers to a wireless group key management system's need to optimize the overhead of communications, computation and key storage during key distribution and updating. Due to the limitations of mobile devices and wireless networks, operational efficiency is the most important issue in wireless group key management. A wireless group key management system cannot be considered deployable in the wireless environment if it fails to satisfy the requirements of operational efficiency. Operational efficiency can be measured in relation to the following three aspects: communication cost, computational cost and key storage cost.

  - Communication cost refers to the number of messages sent by the KDC during the key updating procedure caused by membership changes such as join and leave.
  - Computational cost measures the overhead of processing keying materials on both the KDC and group members' mobile devices during the rekeying. On the KDC, this parameter can be accessed by the number of keys that are encrypted during the rekeying process. For the members' mobile devices, this parameter can be evaluated by the number of received rekeying messages, because group members need to process each received rekeying message to find the right key(s).
  - Key storage cost measures the number of keys stored on both sides of the KDC and group members' mobile devices.

- *Scalability*

Scalability refers to the features of a wireless group key management system that can efficiently address the issues of varying group sizes, widely distributed memberships and highly dynamic membership changes.

- Varying group sizes

  A scalable wireless group key management system is able to deal with group size varying from several dozens to hundreds and thousands. The size of a group should have no or little affect on system performance during key management.

- Widely distributed memberships

  Scalable wireless group key management systems are required to support very large groups where members are widely geographically distributed. In wireless networks, group members are spread across a wireless network domain whose scope may be a metropolis or even a nation. Therefore, scalable wireless group key management systems need to cooperate with underlying wireless networks to support widely distributed groups.

- Highly dynamic membership changes

  Scalable wireless group key management systems are capable of handling a high rate of membership changes. For example, at the beginning of an IPTV broadcast, millions of people may register to join this group. At the end of broadcast, many members leave almost at the same time. A high rate of membership changes should not noticeably decrease the system performance of a scalable wireless group key management system.

- *1-affect-n phenomenon*

The 1-affect-n phenomenon refers to the impact of a single membership change on the remaining members in the whole group. The cause of the 1-affect-n phenomenon is that only one group key or traffic encryption key (TEK) is applied in the group communication. In order to enforce security, the TEK needs to be updated after a single membership change. This TEK updating affects all the remaining members in the group. The 1-affect-n phenomenon can be measured by the number of TEKs employed in the group communication. If there are several TEKs applied in the group communication, the 1-affect-n phenomenon is best restricted within a small area so that it is unable to affect the whole group. Otherwise, the group application suffers from the 1-affect-n phenomenon and consequent performance deterioration.

In addition to performance parameters, other parameters can be applied to evaluate the security features of wireless group key management systems. The security parameters are backward and forward secrecy, prevention of collusion attacks, key independence and trust relationships [Challal & Seba, 2005; Kim & Perrig et al., 2004b].

- *Backward secrecy*

  Backward secrecy refers to the prevention of a new member from being able to decrypt the group communication that it has received before it joins the group. A secure wireless group key management system needs to ensure the backward secrecy.

- *Forward secrecy*

  Forward secrecy requires that a group member who has left a group not be able to access any future group keys, nor should it be able to decrypt any group messages after it leaves the group. This feature also needs to be provided by a secure wireless group key management system.

- *Prevention of collusion attacks*

  Collusion attack refers to a situation where any set of departing members work together to regain the current group key by applying the old keying materials known by them. A secure wireless group key management system should be free from collusion attacks.

- *Key independence*

  Key independence means that all keying materials should be completely independent from each other. Disclosure of any single key does not compromise other keys. A secure wireless group key management should provide this feature to secure keying materials.

- *Trust relationships*

  Trust relationships refer to that a wireless group key management system should not trust any intermediate or third party components. Should intermediate or third party components be trusted, the effective deployment and operation of the key management approach would be compromised. In general, within a wireless group communication system, the network provider and service provider are different entities. The service provider who owns the group key should not trust the network provider and visa versa.

Except for the trust relationships, the relationships among the other security parameters are shown in Figure 3.3.

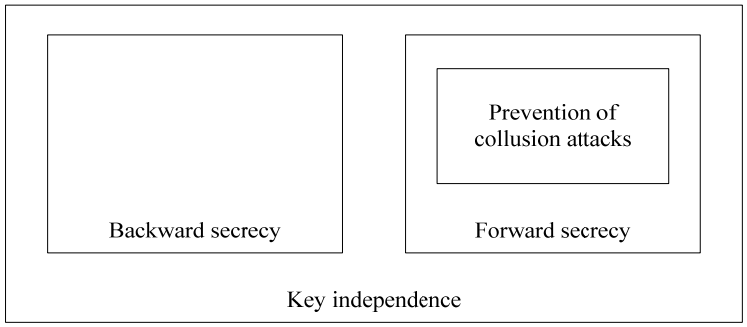


Figure 3.3 Relationships of security properties

The relationships between the security properties are intuitive. 'Forward secrecy' subsumes 'Prevention of collusion attacks' while 'Key independence' subsumes all three other parameters. Key independence is the foundation of the other three security requirements. If keys are generated dependently, the internal relationship among the keys could be exploited by adversaries to gain the current group key, which jeopardizes the forward and backward secrecy. Collusion attack is a special case to challenge the forward secrecy, where a set of departing or evicted members work together to regain access to the current group key by exploiting the relationships between the keys. In order to be immune from collusion attacks, keying materials should be independent.

The last parameter used to evaluate wireless group key management systems is network compatibility. Currently, several types of wireless networks co-exist, including wireless LAN, cellular wireless network and wireless sensor network. Each wireless network has unique properties. Due to these unique properties, no single

group key management system is compatible with all types of wireless networks. A wireless group key management system needs to be especially designed to fit a specific type of wireless network. As mentioned in the section 3.1, we focus our research on the cellular wireless network. The group key management system proposed in this thesis is tailored to the cellular wireless network.

All the above proposed assessment parameters can be used to evaluate wireless group key management systems. However, there is no need to evaluate each component of a wireless group key management system by every parameter. For example, wireless group key management architecture provides an infrastructure for cooperation with the underlying wireless network to address the scalability issue and the 1-affect-n phenomenon. Consequently, wireless group key management architecture can be analyzed by these two parameters. In terms of wireless group key management approaches, operational efficiency is the highest priority due to the limitations of the wireless environment. If a wireless group key management approach cannot meet operational efficiency requirements, it cannot be recognized as a practical approach for the wireless environment. Operational efficiency is therefore the primary criteria to evaluate wireless group key management approaches.

3.2.3 Summary

In this section, we have proposed a formal model for a wireless group key management system. The major contribution of this model is that the underlying wireless network is taken into consideration to address the network-compatible issues of wireless group key management systems. Based on the specific wireless

network infrastructure, group key management architecture and approaches can be developed to fully utilize the capacity of the wireless network to facilitate key management. In addition, three categories of assessment parameters (performance, security and network compatibility) are defined, that can be applied to analyze and evaluate wireless group key management systems. The model and assessment parameters can be considered as a guideline for system designers to design and evaluate wireless group key management systems.

In the remainder of this chapter, this model is applied to the cellular wireless network to develop a relevant wireless group key management architecture.

3.3 A Group Key Management Architecture for the Cellular Wireless Network

Due to the varying characteristics of different wireless networks, group key management needs to be designed for a particular wireless environment. Of all the wireless networks, the cellular wireless network is the most important. First, the cellular network topology is the dominant wireless network structure applied in wireless networks. Almost all the wireless networks (global system for mobile communications (GSM) [Poole, 2006], satellite communications networks and wireless LANs) employ the cellular topology. Moreover, the future wireless broadband systems, WiMax and 3G/4G networks [Ahson & Ilyas, 2007; J.-C. Chen & Zhang, 2004; Chichester & Hoboken, 2008], are also developed based on the cellular network technology. Second, the cellular wireless network is the most widely

implemented and deployed wireless network in the world. For these two reasons, we focus our research on the cellular wireless network. In the following sections, we design a wireless group key management architecture for the cellular wireless network by applying the proposed model.

3.3.1 The Cellular Wireless Network

In order to design an efficient and secure group key management system for wireless networks, designers need to have a profound understanding of the underlying wireless network in order to identify the network-specific features that can be applied to facilitate key management. Hence, in this section, we investigate the characteristics of the cellular wireless network that are useful in designing a wireless group key management system.

In wireless networks, radio spectrum is one of the scarcest available resources, and every effort has to be made to find ways to utilize the spectrum efficiently and to employ network topology that can support as many users as possible. The cellular network topology [MacDonald, 1979] is a major breakthrough in solving the spectrum and system capacity by employing frequency reuse intelligently. Frequency reuse means to spatially reuse the available spectrum so that the same spectrum can support multiple users separated by a distance for the efficient use of the spectrum. In a cellular wireless network, the whole coverage area is divided into a number of small areas, defined as cells. Each cell is served by a low-power transmitter or base station and is provided with a different group of channel frequencies so that all the available bandwidth is assigned to a relative small number of neighboring cells.

Neighboring cells are assigned different groups of frequencies so that interference between the base stations is minimized. This frequency reuse technology is widely applied in the current GSM system [Pahlavan & Krishnamurthy, 2001]. In the 3G/4G system, with the help of CDMA (code division multiple access) technology [Rhee, 1998; Viterbi, 1995], the same spectrum can also be reused in neighboring cells without interference, which further expands the capacity of cellular network [J.-C. Chen & Zhang, 2004; Poole, 2006].

The cellular network structure is shown in Figure 3.4. In a cellular network, users or mobile stations (MSs) connect to the base station (BS), which is managed by the base station controller (BSC). The BSC is connected to the home location register (HLR) and the visit location register (VLR) for network operations such as communication with other wired and wireless networks, registration and maintenance of the connection between the MSs.

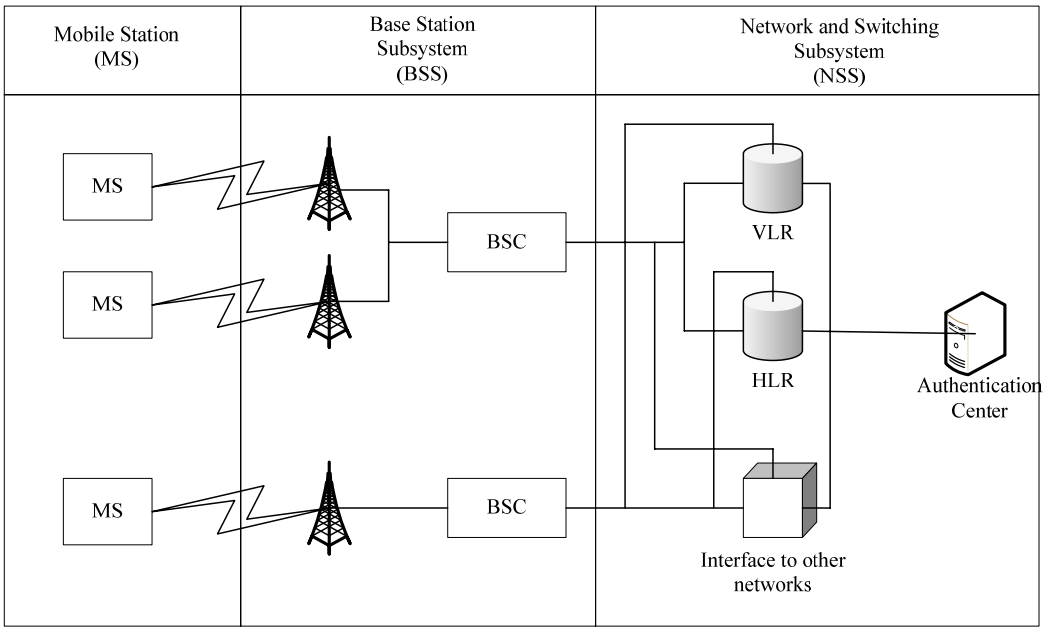


Figure 3.4 The architecture of the cellular wireless network

The most important and unique feature of wireless communication is mobility. This feature provides users with the ability to access the network anywhere and anytime [Akyildiz & McNair et al., 1998; Poole, 2006; Thajchayapong & Peha, 2006]. In the cellular wireless network, a MS is responsible for cell reselection should the user move during the transmission. A MS listens to the broadcast control channel (BCCH) and decides which cell it has to select. The MS measures the received signal strength (RSS) of the current BCCH, compares it to the RSS of the neighboring cell's BCCH and determines to which cell it needs to attach.

In summary, the cellular wireless network has two network-specific features that assist the design of wireless group key management systems. First, the cellular wireless network has a decentralized two-level control structure. The first level consists of base stations and controllers to form distributed base station networks that provide communication services. Each cell in this level is a relatively independent administrative area. HLR and VLR form the second level, the centralized control center, to manage the authentication and routing. Second, the base stations have extensive communication, computation and storage capacity that can be utilized by wireless group key management systems to facilitate key management operations.

3.3.2 A Group Key Management Architecture for Cellular Wireless Network

Based on the two network-specific features of the cellular wireless network, we propose a decentralized wireless group key management architecture. The purpose of this architecture is to solve two performance problems: scalability and the 1-affect-n

phenomenon. In order to provide scalability, the wireless group key management architecture needs to match the two-level structure of the cellular wireless network. Therefore, we adopt a two-tier structure [Hardjono & Cain et al., 2000] to design our wireless group key management architecture (WGKMA). In the WGKMA, the whole group key management domain is divided into a number of smaller administrative areas according to the cellular structure so that each group key management area is also a wireless cell in the cellular wireless network. In each cell, a specific key distribution controller called cell key controller (CKC) is designated to perform key management. This key management architecture is illustrated in Figure 3.5.

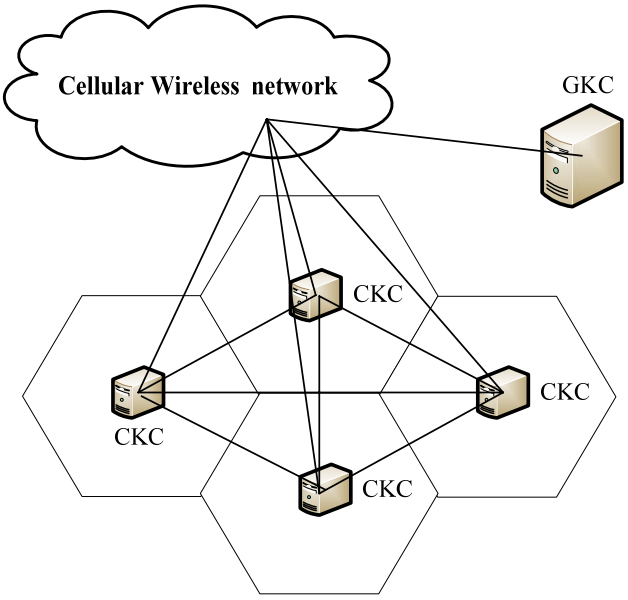


Figure 3.5 Group key management architecture for the cellular wireless network

The purpose of this design is to seamlessly integrate the group key management structure with the underlying wireless network infrastructure to use the capacity of the cellular wireless network to facilitate key management. Based on this key

management architecture, group key management operations are divided into two levels: the cell level and the group level.

- At the cell level, each cell is an independent key management area. Within the cell, the cell key controller (CKC) provided by a wireless network operator is deployed to be responsible for the key management. The key management tasks of the CKC are:
  (i) to apply group key management approaches for the purpose of efficiently performing the key management operations within the scope of the cell, such as distributing group key (traffic encryption key (TEK)) to group members in the cell; and
  (ii) to perform key management for group members who move from one cell to another cell.

Applying the CKC within the cell offers two advantages. First, the application of the CKC provides a universal platform for all secure group applications operating in the cell. Service providers do not need to manage keys within the cell. Leaving them free to focus on key management at the group level. Second, the application of the CKC promotes cost-effectiveness, because service providers do not need to deploy key servers at the cell level. Service providers can instead use the same CKC supplied by the network provider to perform key management within the cell.

- At the group level, a group key controller (GKC) owned by the group service provider is deployed for the purpose of key management within the entire group domain. The major tasks of GKC are:

(i) to authenticate users when users join the group;

(ii) to generate group key (TEK) for the group communication; and

(iii) to distribute group key (TEK) and other control messages to all the CKCs.

In order to efficiently distribute keying materials and control messages between these two key management levels, we define two key-management-related groups: the all-KC group and the cell-control group, depicted in Figure 3.6.

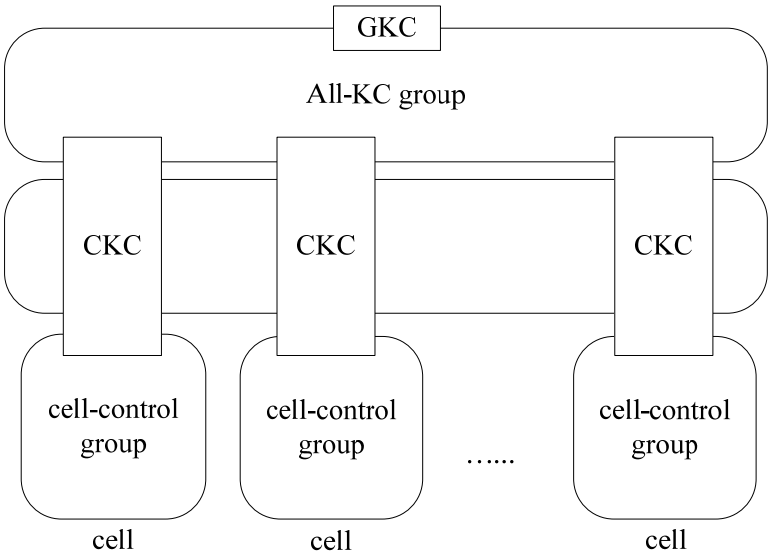


Figure 3.6 Two key-management-related groups

- At the group level, a group called the all-KC-group is defined. This is an administrative group and consists of the GKC and all the CKCs. The all-KC-group has a fixed global multicast address associated with the group application and is initiated by the GKC. This group exists as long as the group application operates. In the group initiation stage, only the GKC is in the group. When a user joins the group application, the corresponding CKC joins this group to receive group key from the GKC. The GKC uses this group to distribute the

new group key to the CKCs. All the controlling messages are distributed from the GKC to the CKCs via this group as well.

- At the cell level, we define another key-management-related group called the cell-control-group. The operational scope of this group is restricted to the cell. Each cell has an instance of this group and this group is associated with a group application and exists as long as there are group members in the cell. In the initiation stage, the CKC creates an instance of the cell-control-group when a user in the cell joins the group application. Once a group application ceases to have any members within the cell, the CKC terminates the corresponding cell-control-group. In order to efficiently distribute keying materials to the legitimate group members within the cell, the CKC applies a group key management approach within the cell-control-group. This is discussed in detail in the next chapter.

In order to minimize the impact of the 1-affect-n phenomenon, two levels of TEKs are applied: the group TEK (GTEK) and the cell TEK (CTEK).

- GTEK is operated at the group level and is owned by the GKC. GTEK is used to encrypt group data by the sender(s) during the group communication. GTEK does not required immediate updating when membership change occurs, because CTEK can be updated to respond automatically to the membership change. GTEK can be updated periodically or according to the group key management policy. GTEK is delivered to the group members from the GKC directly when users register to join the group.

- CTEK is operated at the cell level and is owned by the CKC. In WGKMA, each cell is an independent key management area and is able to apply its own TEK. CTEK is generated independently by each CKC and is updated once membership change occurs in the cell. Therefore, multiple CTEKs are applied at the cell level. User can receive the CTEK from the CKC when it registers to receive group communication via IGMP [Cain & Deering et al., 2002; Fenner, 1997].

In order to reduce the impact of the 1-affect-n phenomenon, dual encryption is applied in WGKMA. During the group communication, the sender encrypts the group data with the GTEK at the application layer and sends the data to the whole group via the wireless network. Before the base station multicasts the group data within its cell, the base station encrypts the data again with the CTEK at the network layer or lower. The purpose of encryption at the lower layer is to reduce interference in the group communication. Each group member knows two TEKs: GTEK and CTEK. When group data is received by a member, first, the member decrypts the received data with the CTEK to obtain the encrypted group data, and then decrypts it using the GTEK to access the group communication content. Table 3.1 presents the number of TEKs held by each entity in the proposed wireless group key management architecture.

Table 3.1 The number of TEKs held by entities

Involved entities	Number of keys	Name of key(s)
GKC	1	Group TEK
Base station	1	Cell TEK
Sender(s)	1	Group TEK
Receivers	2	Group TEK and Cell TEK

3.3.3 Performance Analysis of the Wireless Group Key Management Architecture

The purpose of the proposed wireless group key management architecture is to tackle the performance issues of scalability and the 1-affect-n phenomenon. In this section, we analyze the wireless group key management architecture (WGKMA) with respect to these two assessment parameters.

- Scalability

The scalability of a group key management system refers to a system's ability to cope with various group sizes, handle widely geographically distributed group members and deal with highly dynamic membership changes. The proposed wireless group key management architecture has this functionality.

First, the proposed architecture is based on the cellular network topology. Each wireless cell is also a group key administrative area. This helps the proposed wireless group key management architecture seamlessly integrate with the underlying cellular wireless network. This integration makes the key management system utilize the

infrastructure of the cellular wireless network to cover a large geographical area. The scalability of the group key management system can be increased in line with the expansion of the cellular wireless network when new base stations are built.

Second, instead of key management being processed by a centralized KDC, in WGKMA, a distributed key management approach is applied. Key management within a cell can be operated by each CKC independently and key management in the whole wireless domain can be performed by the CKCs in parallel. This distributed processing increases the scalability of the group key management system to deal with a large number of group members and highly dynamic membership changes.

- 1-affect-n phenomenon

In WGKMA, each cell has it own CTEK and multiple CTEKs are applied in the system to minimize the impact of the 1-affect-n phenomenon. When a membership change occurs in a cell, the affected CTEK is updated immediately to respond to the membership change while the GTEK does not need a corresponding update due to the application of two-level TEKs. In addition, the operational scope of the affected CTEK restricts the updating to the affected cell so that the remaining wireless cells do not need to refresh their CTEKs. Rekeying the entire group is unnecessary, and the group communication is not altered by the rekeying of the affected cell.

The features of this architecture reduce the impact of the 1-affect-n phenomenon and improve the performance of group key management. In addition, the architecture also maintains the quality of service (QoS) of group applications, especially for some high security group applications such as military communications where the group

communication has to be interrupted during key updating. In conclusion, by applying multiple TEKs, the 1-affect-n phenomenon is able to be minimized in WGKMA.

3.4 Summary

In this chapter, a formal wireless group key management model has been proposed to illustrate group key management in the wireless environment. The major contribution of this model is to take the underlying wireless network into consideration. When the wireless network is taken into account, the performance of wireless group key management benefits. This chapter has also given the definition of evaluation parameters based on this model for the purpose of analysis and evaluation of wireless group key management systems.

By applying the proposed model to the cellular wireless network, a wireless group key management architecture (WGKMA) has been developed. This architecture addresses the performance problems of scalability and the 1-affect-n phenomenon with the following features.

- WGKMA provides scalability by seamlessly integrating the group key management structure with the underlying cellular wireless network. In WGKMA, the whole key management domain is divided into a number of small key administrative areas that are also a wireless cells in the cellular wireless networks. Key management can therefore be operated independently in each cell and in parallel within the cellular wireless network.

- WGKMA applies multiple TEKs to minimize the 1-affect-n phenomenon. The group key management operations are divided into two levels: the group level and the cell level. Each level is associated with a TEK. Each cell has its own CTEK to tackle the 1-affect-n problem.

Although the proposed wireless group key management architecture has addressed two performance issues, the proposed architecture only provides a infrastructure and efficient group key management approaches are still required to be performed within the cell. In the next chapter, we propose a group key management approach that is tailored to be efficiently operated in the cellular wireless network.

Chapter 4

Hybrid Group Key Management

4.1 Introduction

How to efficiently distribute keying materials to a number of group members is the most important and challenging question in group key management. A key controller needs to follow a proper algorithm to efficiently perform key management operations such as key generation, key distribution and key updating. Operational efficiency is therefore the top priority of group key management. This becomes even more critical when a group key management approach is applied in the wireless environment due to the resource limitations of both wireless networks and mobile devices.

In Chapter 3, we have proposed a wireless group key management network architecture for the cellular wireless network. Although this architecture tackles two performance problems, scalability and the 1-affect-n phenomenon, it still needs a group key management approach to be operated within the cell to efficiently manage the keying materials for group members.

In Chapter 2, several group key management approaches have been investigated. However, these encounter serious operational efficiency problems when they are applied in the wireless cell based on the proposed architecture. In this chapter, we propose a new wireless group key management approach: hybrid group key management (HGKM) [Y. Chen & Wang et al., 2006; Wang & Damodaran et al., 2006; Wang & Le, 2007a; Wang & Le et al., 2007], that is dedicated to the cellular wireless network based on the proposed wireless group key management architecture. This new group key management approach, HGKM, is designed to address the operational efficiency issues through the following features:

- a small key management structure that enables micro-key management operations to achieve efficiency for both the key controller and group members;
- a combination of centralized and decentralized key management approaches that allows members to involve in key management in order to reduce the overhead for the key controller during rekeying;
- a simple message delivery scheme that supports reliable transmission of rekeying messages; and
- key controllers that co-operate to facilitate rekeying during the handoff.

The remainder of this chapter is organized as follows. In section 4.2, we discuss the inefficiency of existing group key management approaches when they are applied in the wireless environment. In section 4.3, we investigate the proposed HGKM to illustrate its operation in terms of the join, leave and handoff actions. In section 4.4, we analyze and evaluate the operational efficiency of the proposed HGKM on the parameters of communication cost, computation cost and key storage cost. Finally,

section 4.5 summarizes the chapter.

4.2 Existing Group Key Management Approaches

In Chapter 2, we have investigated two categories of existing group key management schemes: distributed and centralized approaches. In the distributed group key management approaches such as Group Diffie-Hellman Key Exchange [Steiner & Tsudik et al., 1996, 2000] and Tree-based DH Key Management [Kim & Perrig et al., 2004b], there is no explicit key controller that manages the group key and supporting keys. The generation of the group key is done by all group members in a contributory way; that is, each member contributes its own parts to calculate the shared group key. A number of expensive exponential computations are required in the key generation. When it is applied to the cellular wireless network, the distributed approach encounters operational efficiency problems, as most mobile devices cannot afford a large number of expensive exponential computations. Furthermore, in order to calculate the group key, each member needs to broadcast its contribution within the group following a one-by-one transmission pattern. Completing the contribution broadcast in a large group becomes a lengthy process. Distributed group key management approaches consequently have a slow key converging speed during key generation.

In summary, two serious performance issues - the expensive computation requirement and the slow speed of key generation - prevent the distributed key management approaches from being applied to the cellular wireless network.

The centralized group key management schemes such as LKH [Wallner & Harder et al., 1999; Wong & Gouda et al., 2000] and OFT [McGrew & Sherman, 1998] also face operational efficiency problems when they are deployed in the cellular wireless network. First, current centralized group key management approaches suffer the 1-affect-n phenomenon. In centralized key management, all group members are organized into a single hierarchical key tree. Once a membership change occurs, the rekeying affects all remaining members in the key tree. Second, centralized key management suffers from the high maintenance costs of a key tree. In order to perform key management efficiently, a hierarchical key tree needs to be as balanced as possible. However, in a large and highly dynamic group, maintaining a key tree's balance is difficult. It is also costly in terms of system resources. Finally, in order to achieve efficient transmissions, multicast transmission is widely used to distribute keying materials. In a centralized group key management approach, all rekeying messages are multicasted within the group and received by every member, and each group member has to process all the received rekeying messages in order to determine if it is the intended recipient. However, not every rekeying message is relevant to all group members (discussed in section 2.2). From the member's perspective, this rekeying mechanism is not efficient, because resources are wasted during the processing of irrelevant rekeying messages. Moreover, in a highly dynamic group, the frequent rekeying message process may overwhelm the limited capacity of handheld devices.

In summary, because of operational efficiency issues, distributed and centralized group key management approaches are not suitable for application in the cellular

wireless network. A more efficient group key management approach is required for the cellular wireless network. In the next section, we propose the HGKM approach, which is designed to be deployed within the wireless cells of the cellular wireless network.

4.3 Hybrid Group Key Management (HGKM)

In this section, we delineate the proposed HGKM approach in detail. This section is divided into two parts. In the first part, from section 4.3.1 to section 4.3.6, we investigate the structure and key management operations of HGKM. In the second part, from sections 4.3.7 to section 4.3.8, we discuss two features of HGKM: reliable message delivery and optimization of key management structure.

4.3.1 Logical Key Management Structure in HGKM

From our analysis of centralized group key management approaches in section 2.2, it has become apparent that the reason for operational inefficiency in centralized approaches is rooted in the organization of all group members in one single and large hierarchical structure. In a hierarchical key tree, any membership change affects all the members in the key tree. With a large and dynamic group, maintaining the balance of the key tree is costly. In order to minimize the rekeying impact on the group members and to reduce the overhead of key management, we propose that micro-key management operations be performed on small management structures called *operation units*. An operation unit is a small fixed-sized key management

structure. Each operation unit contains a hierarchical structure for the purpose of key management. In HGKM, all the key management operations are performed based on the operation units.

Under the HGKM approach, all group members within a cell form a key management group called cell-key-management-group for the purpose of key distribution. The cell-key-management-group is composed of operation units. Members are assigned into one and only one operation unit when they join the group. There are two types of operation units in HGKM: *leader unit* and *member unit.* These two kinds of operation units form a two-level logical structure for the purpose of key management, as shown in Figure 4.1.

The upper level, called the leader units level, is formed by leader unit(s). There are two roles in the leader unit: *leader* and *leadership candidate*. Leader refers to a group member in a leader unit who is designated as a leader of a member unit in the lower level. The responsibility of the leader is to assist the cell key controller (CKC) to distribute the keying materials to the group members within its member unit. On the other hand, leadership candidate refers to a member in a leader unit who has not been assigned as a leader yet. Its responsibility is to work as a backup to the current leader. When a leader leaves the group, a candidate can be designated as the new leader of the affected member unit immediately. The purpose of having leadership candidates is to reduce the operational cost of rekeying when the leader leaves the group. This is further discussed in section 4.3.4.

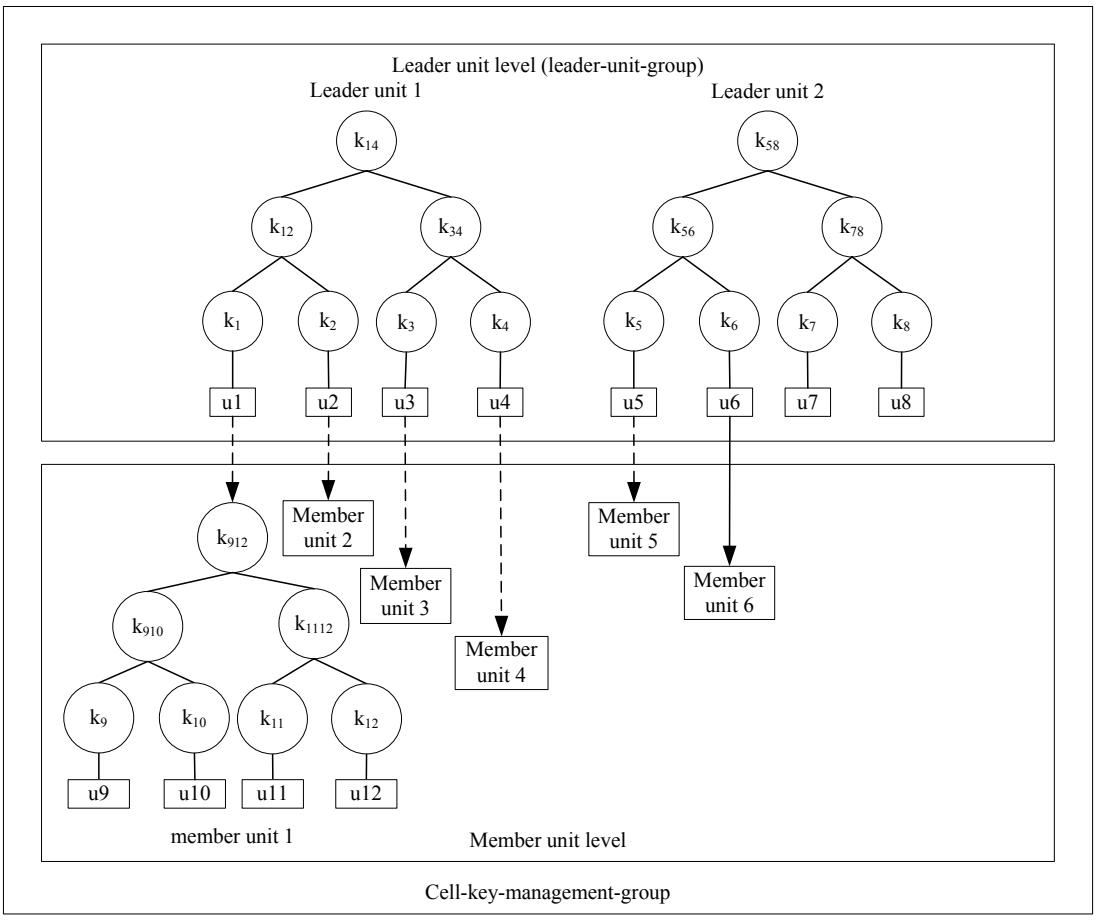


Figure 4.1 Logical structure of HGKM

Within a leader unit, all the members are arranged into a key tree that is the same as that in the LKH. In this key tree, each leaf node is associated with a leader or a leadership candidate. Each intermediate node in the key tree represents a supporting key and the key associated with the root node represents the unit key (i.e. k_{unit}). The unit key is shared by all members in the leader unit for the purpose of key distribution within the unit. Each member needs to store a set of keys along the path from its leaf node to the root. All the leader units are controlled by the cell key controller (CKC) and form another key management related group called *leader-unit-group* for the purpose of key distribution at the leader unit level.

The lower level is the member unit level, which is comprised of member units. In this level, group members are assigned into member units. Members in each member unit also form a key tree for efficient key distribution. In the key tree, each member is associated with a leaf node and the root node represents the unit key of that member unit. The difference between leader units and member units is that each member unit has a designated leader from a leader unit.

In summary, the logical key management structure of HGKM has the following features.

- The logical key management structure is composed of small operation units. Each unit is an operation structure where key management operations can be performed.
- The logical key management structure combines the features of the centralized and distributed approaches to allow members to participate in the key management activities in order to reduce the overhead of key management.
- An operation unit, which is a small fixed-sized key management structure, applies a hierarchical structure to facilitate key management within the unit.

4.3.2 IP Multicast Addressing in HGKM

IP multicast transmission is widely applied during rekeying in order to improve communication efficiency on the CKC. In HGKM, each operation unit is bound with an IP multicast address. A tuple (S, C) is applied to identify an operation unit, where S is the IP multicast address of the secure group application and C is the assigned local multicast address to the operation unit within the cell. Therefore, (S, C) and (S’,

C) are two different operation units. A member in the operation unit (S, C) cannot automatically receive the messages sent to the operation unit (S', C). A total of 2^{24} class D address (232.0.0.1 – 232.255.255.255) allocated by IANA [Albanna & Almeroth et al., 2001] can be used as C in the wireless cell, as shown in Figure 4.2. Each secure group application can thus have up to 16 million (2^{24}) local IP multicast addresses to allocate to one cell.

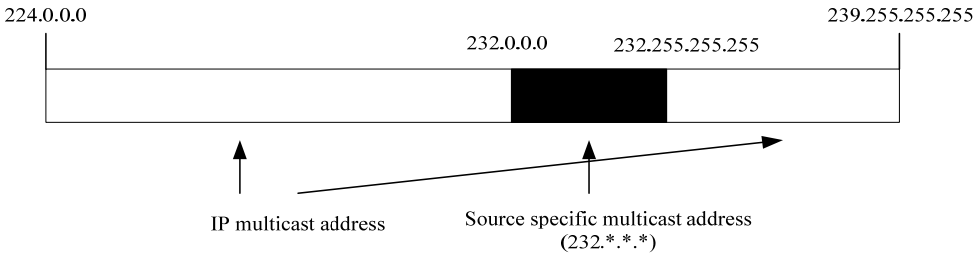


Figure 4.2 Source specific IP multicast address

Besides the operation units, other two types of key management related groups also need to be assigned IP multicast addresses:

- the cell-key-management-group comprising the operation units; and
- the leader-unit-group comprising the leader units.

In HGKM, each member simultaneously belongs to the several key management groups for the purpose of key management. A leader is a member of the following three key management groups: the cell-key-management-group, the leader-unit-group and the leader unit. Meanwhile, a common member (a member who is not a leader) is a member of both the cell-key-management-group and the member unit. Each member thus needs to listen to multiple IP addresses to receive keying materials.

4.3.3 Generation of Leader and Member Units

In the initial stage of HGKM, a CKC first creates the leader unit. To ensure a new leader is available as soon as a leader leaves the group, there also needs to be leadership candidates in the leader unit. The ratio of leadership candidates varies from application to application. With respect to a highly dynamic group, the ratio of leadership candidates is high to minimize the impact of a leader's departure. On the other hand, for a stable group application, the ratio is low because few leaders leave the group during the group communication.

After the CKC completes the generation of leader unit(s), the CKC creates member units for the new incoming group members and designates a member from the leader unit to be the leader of the newly-generated member unit.

During the generation of operation units, the CKC continually monitors the changes in the ratio of leadership candidates. If the ratio falls below the threshold predefined in the key management policy, the CKC creates leader unit(s). Otherwise, the CKC generates member units for the new incoming group members.

4.3.4 The Join Operation

Based on the proposed wireless group key management architecture described in the section 3.3, when a user joins a group, the CKC needs to enforce backward secrecy to prevent the new joining member from decrypting previous group data by updating the cell traffic encryption key (CTEK). Therefore, in HGKM, the join operation starts with the user sending the join request to the group key controller

(GKC) for authentication and authorization. In addition, the incoming member also sends a request (via Internet Group Management Protocol (IGMP) [Cain & Deering et al., 2002; Fenner, 1997]) to the CKC requesting group communication data from the base station.

$$\text{user} \rightarrow \text{GKC: } \{\text{group join request}\}$$

$$\text{user} \rightarrow \text{CKC: } \{\text{request for receiving the group communication data}\}$$

After receiving the join request, the GKC validates the user. If authentication is successful, the GKC sends the new incoming member the group traffic encryption key (GTEK) encrypted with a pair-wise key known only to the GKC and the user, while the GKC informs the CKC that the user is authenticated and authorized.

$$\text{GKC} \rightarrow \text{user: } \{k_{\text{GTEK}}\}k_{\text{GKC-user}}$$

$$\text{GKS} \rightarrow \text{CKC: } \{\text{user is authenticated}\}$$

After receiving this control message from the GKC, the CKC contacts the new joining member and invokes the join procedure. There are three steps to perform the join action in the cell.

- Step1: the CKC assigns the new joining member into an operation unit, where the leader unit has priority over the member unit.
- Step 2: the CKC sends the new member a set of keys including the cell traffic encryption key (CTEK) and the supporting keys it is entitled to know.
- Step 3: the CKC invokes the join rekeying procedure to update the keys for the remaining members within the cell.

In HGKM, there are two types of join operations within the cell: the joining leader unit and the joining member unit, as illustrated in Figure 4.3.

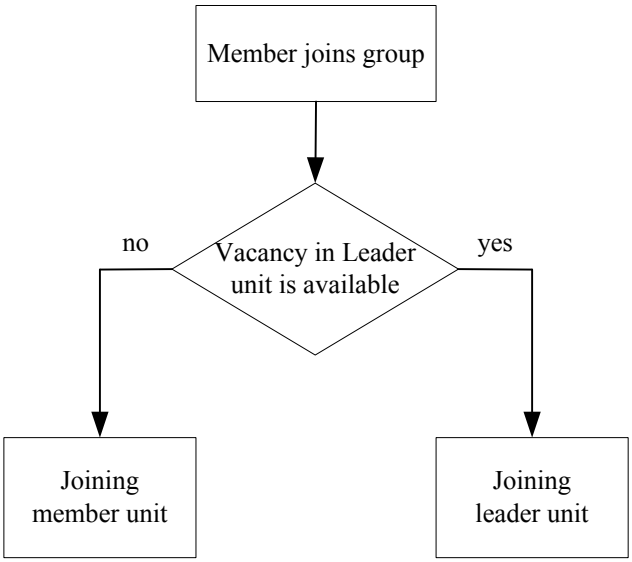


Figure 4.3 Two kinds of join operations

The CKC assigns the new member into an operation unit according to the key management policy, where the leader unit is given priority over the member unit. The purpose of doing this is to ensure there are enough members working as leaders and leadership candidates in the leader units' level for key management, because the responsibility of a leader is to assist the CKC to distribute rekeying messages within its member unit. In the following sections, we investigate the rekeying operations for the above two kinds of join actions.

4.3.4.1 The Joining Leader Unit

Once the CKC finds an available empty slot in the leader unit, the CKC assigns the joining member into that slot. In order to ensure backward secrecy, the CKC needs to generate new keys to replace the affected current keys. The affected keys are

the keys on the key tree from the leaf node of the new incoming member along the path to the root node. After sending the newly-generated keys to the new member, the CKC invokes the rekeying procedure to update the affected keys for the remaining group members. This rekeying process follows the "bottom-to-top" method and is divided into two steps.

- Step 1: the CKC updates the keys for the directly-affected leader unit where the new member resides. This rekeying procedure is the same as that in LKH (described in section 2.2.1).
- Step 2: the CKC multicasts a rekeying message in the cell-key-management group where all remaining members reside to update the CTEK for the group members within the cell.

In order to better illustrate the joining leader unit, we provide an example, shown in Figure 4.4, to further describe this rekeying procedure.

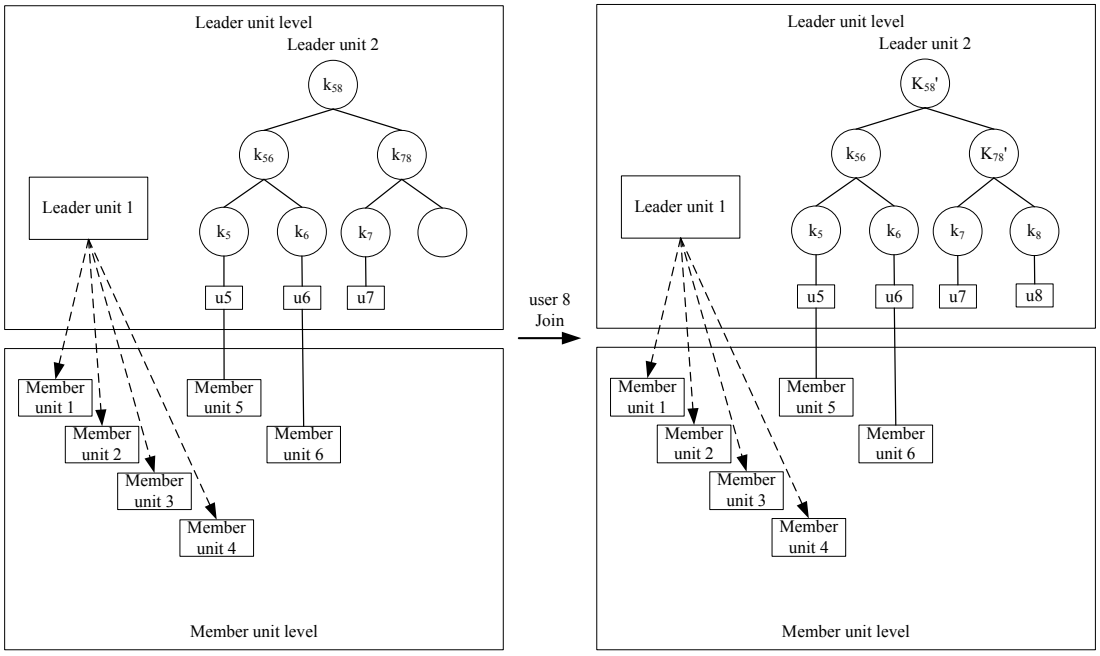


Figure 4.4 Joining a leader unit

In Figure 4.4, when user 8 joins the group, the CKC assigns user 8 into leader unit 2 as a leadership candidate. The CKC generates the new keys $(k_{\text{CTEK}}', k_{78}', k_{58}')$ to replace the current keys $(k_{\text{CTEK}}, k_{78}, k_{58})$ to ensure backward secrecy. The CKC sends these new keys to user 8.

$$\text{CKC} \rightarrow \text{user 8: } \{k_{\text{CTEK}}', k_{78}', k_{58}'\}k_8$$

In this rekeying message, the total of $h_{\text{unit}} + 1$ keys is encrypted by the CKC, where h_{unit} is the height of the key tree for the operation unit.

After rekeying the new joining user, the CKC invokes the rekeying procedure to update the keys for the remaining group members within the cell. Following the rekeying procedure, in step 1, the CKC updates the affected keys in leader unit 2 where the new member, user 8, has been assigned. The CKC generates two rekeying messages for user 7 and users 5 and 6 respectively to update the affected keys.

$$\text{user 7: } \{k_{78}', k_{58}'\}k_7$$
$$\text{user 5,6: } \{k_{58}'\}k_{56}$$

Due to the small size of these rekeying messages, the CKC places them into a single integrated rekeying message and multicasts it directly to the affected leader unit 2.

$$\text{CKC} \rightarrow \{\text{leader unit 2}\}\text{: } \{\text{for } \{\text{user 7}\}\text{: } \{k_{78}', k_{58}'\}k_7,$$
$$\text{for } \{\text{user 5,6}\}\text{: } \{k_{58}'\}k_{56}\}$$

When the members in leader unit 2 receive this integrated rekeying message, each member can gain the latest keys from the corresponding section.

In this step, we can observe that the CKC sends a single rekeying message and the number of keys encrypted is:

$$1 + 2 + \cdots + h_{\text{unit}} = \frac{h_{\text{unit}}(h_{\text{unit}} + 1)}{2}$$

where h_{unit} is the height of the key tree for the operation unit.

After updating keys in leader unit 2, in step 2, the CKC generates a rekeying message that contains the latest cell TEK (k_{CTEK}') encrypted by the current CTEK (k_{CTEK}) and multicasts it within the cell-key-management-group that accommodates all the group members within the cell.

$$\text{CKC} \Rightarrow \{\text{cell-key-management-group}\}: \{k_{\text{CTEK}}'\}k_{\text{CTEK}}$$

When the remaining group members receive this message, they can gain the new CTEK for the future group communication.

During the rekeying procedure of the joining leader unit, it can be observed that the CKC sends 3 rekeying messages. One is for the new joining member, one is for the directly-affected leader unit where the new member resides and the last is for all remaining members. In these three rekeying messages, the total number of keys encrypted by the CKC is

$$(h_{\text{unit}}+1)+(1+2+\cdots+h_{\text{unit}})+1=\frac{(h_{\text{unit}}+1)(h_{\text{unit}}+2)}{2}+1$$

where h_{unit} is the height of the key tree for the operation unit.

In relation to the group members, during the rekeying, the members in the directly-affected leader unit receive 2 rekeying messages; the first is the integrated message containing all the keying materials for leader unit 2 while the other contains the new CTEK for all remaining group members. The members outside the directly-affected leader unit only receive one single rekeying message to update the CTEK.

4.3.4.2 The Joining Member Unit

If the CKC cannot find an available slot in the leader unit(s) for the new joining member, then the incoming user is assigned into a member unit. The join procedure is similar to the joining leader unit. The CKC generates new keys to replace the affected current keys in the directly-affected member unit and sends these new keys to the new joining user. After this, the CKC invokes the rekeying procedure to update keys for the remaining group members. Two steps (similar to those performed for the joining leader unit) are involved.

- Step 1: the CKC updates the keys for the directly-affected member unit where the new joining member has been assigned. The CKC generates and multicasts a integrated rekeying message which contains all the rekeying messages required by the directly-affected members unit.
- Step 2: the CKC multicasts a rekeying message that contains the new CTEK encrypted by the current CTEK to the cell-key-management group to update CTEK for all remaining group members.

We give an example, shown in Figure 4.5, to further illustrate this rekeying procedure. When user 9 in Figure 4.5 joins the group, the CKC assigns it into member unit 2. The CKC generates the new keys $(k_{\text{CTEK}}', k_{910}', k_{912}')$ and sends them to new joining member, user 9.

$$\text{CKC} \rightarrow \text{user 9: } \{k_{\text{CTEK}}', k_{910}', k_{912}'\}k_9$$

In this rekeying message, it can be observed that the number of keys encrypted by the CKC is $h_{\text{unit}} + 1$, where h_{unit} is the height of the key tree for the operation unit.

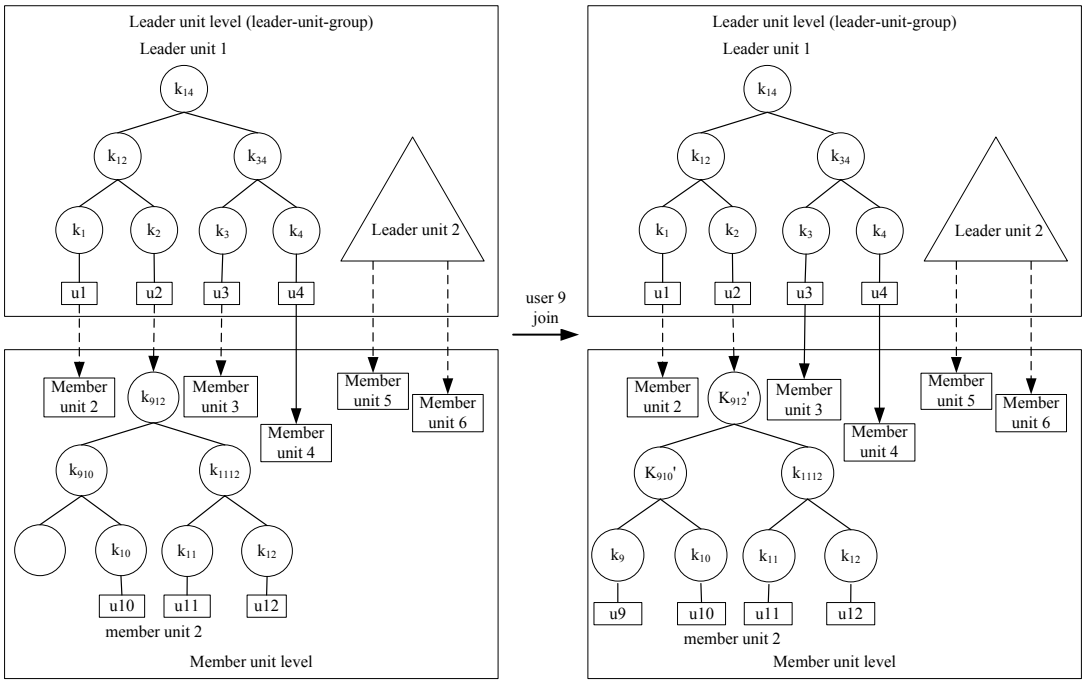


Figure 4.5 Joining a member unit

After rekeying user 9, the CKC invokes the rekeying procedure to update the keys for the remaining group members. First, in step 1, the CKC updates the keys for the directly-affected operation unit, member unit 2. The CKC refreshes the current keys for users 10, 11, 12 and user 2 (who is the leader of member unit 2) by multicasting a integrated rekeying message.

$$\text{CKC} \rightarrow \{\text{member unit 2}\}: \{\text{for } \{\text{user 10}\}: \{k_{910}', k_{912}'\}k_{10}$$
$$\text{for } \{\text{user 11,12}\}: \{k_{912}'\}k_{1112}$$
$$\text{for } \{\text{user 2}\}: \{k_{912}'\}k_2\}$$

After receiving this message, users 2, 10, 11, 12 can update their keys from the corresponding section. For the purpose of reliable delivery (discussed later in this section), the leader of member unit 2, user 2, stores this rekeying message for a period of time until it receives new keying materials from the CKC. In this process, the CKC sends just one rekeying message and the number of keys encrypted by the

CKC during this step is:

$$1+1+2+\cdots+h_{\text{unit}}=\frac{h_{\text{unit}}(h_{\text{unit}}+1)}{2}+1$$

where h_{unit} is the height of the key tree for the operation unit.

After rekeying the directly-affected member unit, in step 2, the CKC multicasts a rekeying message, which contains the new CTEK, k_{CTEK}', to the cell-key-management-group to update the CTEK for all remaining group members within the cell.

$$\text{CKC} \Rightarrow \{\text{cell-key-management-group}\}:\{k_{\text{CTEK}}'\}k_{\text{CTEK}}$$

From this example, it can be observe that three rekeying messages are sent by the CKC during the whole rekeying process. One message is for the new joining group member, one is for the directly-affected member unit and the finial message is for the remaining members. The number of keys encrypted by the CKC is:

$$(h_{\text{unit}}+1)+(1+1+2+\cdots+h_{\text{unit}})+1=\frac{(h_{\text{unit}}+1)(h_{\text{unit}}+2)}{2}+2$$

where h_{unit} is the height of the key tree for the operation unit.

During this rekeying, the leader and the members of the directly-affected member unit receive two rekeying messages: the integrated rekeying message and a message updating the CTEK for all remaining group members. The group members outside the directly-affected member unit only receive one rekeying message updating the CTEK.

4.3.4.3 Summary

Based on the rekeying process for the join operation, it can be observed that the key updating is restricted to the directly-affected operation unit. Therefore, the key

management in HGKM is called *micro-key management.* The small size of the operation unit ensures a low operational join cost for HGKM (the operational efficiency of HGKM is discussed in detail in section 4.4.). Moreover, during the rekeying, the CKC applies one single integrated message to contain all the rekeying information for the directly-affected operation unit. This method utilizes the capacity of multicast transmission and further reduces the communication cost incurred during the rekeying. Table 4.1 summarizes the number of messages sent by the CKC and the number of keys encrypted by the CKC during the rekeying process in two different types of join operations.

Table 4.1 The operational costs of the join operation for the CKC in HGKM

	Number of messages sent out by CKC	Number of keys encrypted by CKC
Joining leader unit	3	$\frac{(h_{\text{unit}}+1)(h_{\text{unit}}+2)}{2}+1$
Joining member unit	3	$\frac{(h_{\text{unit}}+1)(h_{\text{unit}}+2)}{2}+2$

h_{unit} : the height of the key tree for operation unit in HGKM

4.3.5 The Leave Operation

When a member leaves the group, the CKC must update the keys known by the departing member to ensure forward secrecy to prevent the departing member from accessing future group communication. The leave operation can either be invoked by a user through sending a leave request or initiated by the GKC to evict a user. Similar to the join process, there are two types of leave: leaving a member unit and leaving a leader unit.

4.3.5.1 The Leaving Member Unit

When a member leaves a member unit, the CKC needs to replace the current CTEK and supporting keys known to the departing member to enforce forward secrecy. After generating new keys, the CKC invokes the rekeying procedure to update keys for the remaining group members following a "bottom to-top" approach. There are two steps in this rekeying process.

- Step 1: the CKC generates rekeying messages for the directly-affected member unit the member leaves. Similar to the join operation, all rekeying messages can be placed in a integrated message for transmission.
- Step 2: the CKC generates rekeying messages for the leader units to update the CTEK for them. In order to improve communication efficiency, all these rekeying messages can also be placed in one integrated message. After receiving this integrated message, the leaders can gain the latest CTEK and distribute this new CTEK within their own member units on behalf of the CKC.

To better understand this rekeying procedure, we present an example shown in Figure 4.6 to illustrate the rekeying involved in leaving a member unit.

For user 9 in Figure 4.6 to leave the group, the keys $(k_{\text{CTEK}}, k_{910}, k_{912})$ need to be updated. After generating the new keys $(k_{\text{CTEK}}', k_{910}', k_{912}')$, the CKC invokes the rekeying procedure to update the keys for the remaining group members.

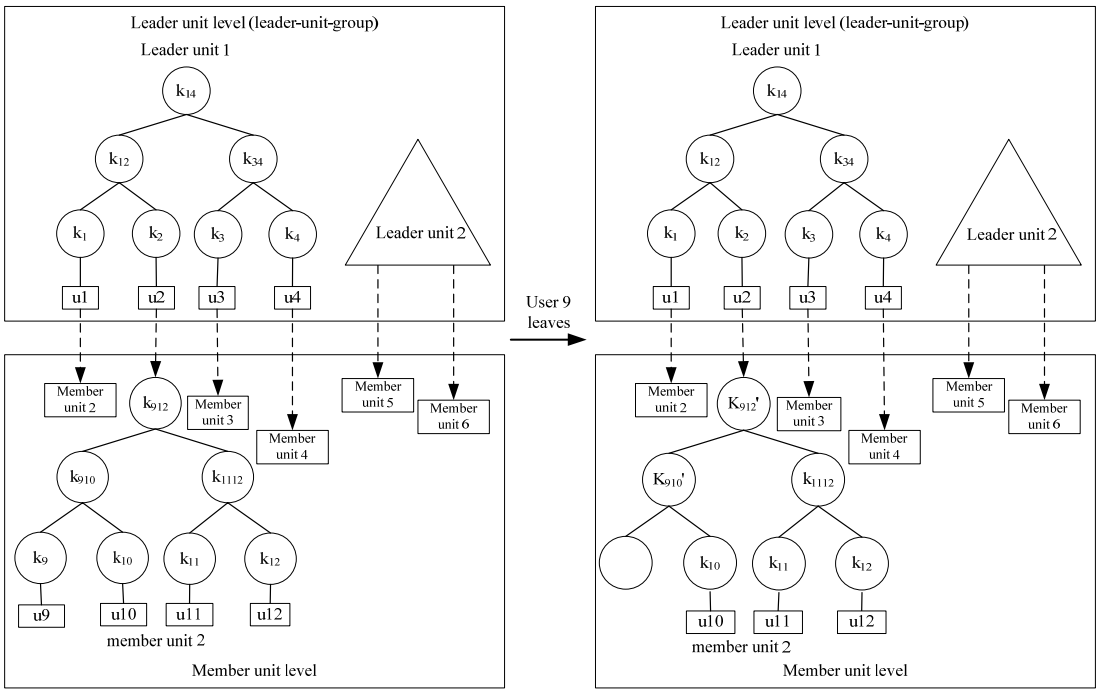


Figure 4.6 Leaving a member unit

First, in step 1, the CKC generates rekeying messages to update the keys in the directly-affected operation unit, member unit 2, where the member leaves.

user 10: $\{k_{910}, 'k_{912}'\}k_{10}$

user 11,12: $\{k_{912}'\}k_{1112}$,

user 2: $\{k_{912}'\}k_2$

In order to improve network communication efficiency, the CKC places these rekeying messages into an integrated message and sends it directly to the affected member unit 2.

$$\text{CKC} \rightarrow \{\text{member unit 2}\}: \{\text{for } \{\text{user 10}\}: \{k_{910}, 'k_{912}'\}k_{10},$$
$$\text{for } \{\text{user 11,12}\}: \{k_{912}'\}k_{1112},$$
$$\text{for } \{\text{user 2}\}: \{k_{912}'\}k_2\}$$

When users 10, 11, 12 and the unit leader (user 2) receive this integrated rekeying message, they collect the useful rekeying message from the corresponding sections in

order to update their own keys affected by the departure of user 9. Similar to the join procedure, for the purpose of reliable delivery (discussed later in this section), the unit leader, user 2, stores this integrated rekeying message for a period of time until it receives replacement keying materials from the CKC.

In step 1, we can observe that the CKC sends only one single rekeying message that contains all the rekeying information for the directly-affected member unit. The number of keys encrypted by the CKC during this step is:

$$1+1+2+\cdots+h_{\text{unit}}=\frac{h_{\text{unit}}(h_{\text{unit}}+1)}{2}+1$$

where h_{unit} is the height of the key tree for the operation unit

In step 2, the CKC generates rekeying messages for the leader units to update the CTEK for them. All these rekeying messages can also be placed in one integrated message that is multicasted to the leader units via the leader-unit-group that contains all the members in the leader units.

$$\text{CKC} \Rightarrow \text{leader-unit-group: } \{\{k_{\text{CTEK}}'\}k_{\text{leader\_unit\_1}}, \{k_{\text{CTEK}}'\}k_{\text{leader\_unit\_2}}\}$$

After receiving this integrated message, the leaders pick up the useful rekeying message from the corresponding sections and update their CTEK. The leaders then distribute this latest CTEK within their member units.

$$\text{user 1} \Rightarrow \text{member unit 1: } \{k_{\text{CTEK}}'\}k_{\text{member\_unit\_1}}$$

$$\text{user 2} \Rightarrow \text{member unit 3: } \{k_{\text{CTEK}}'\}k_{912'}$$

$$\text{user 3} \Rightarrow \text{member unit 3: } \{k_{\text{CTEK}}'\}k_{\text{member\_unit\_3}}$$

$$\text{user 4} \Rightarrow \text{member unit 4: } \{k_{\text{CTEK}}'\}k_{\text{member\_unit\_4}}$$

In this step, we can observe that the CKC still sends out one single rekeying message for the CTEK updating. The number of keys encrypted by the CKC in this step is the

number of leader units (i.e. $n_{\text{leader\_units}}$).

From this example, it can be observed that the CKC needs to send out two integrated rekeying messages during the rekeying: one for the directly-affected member unit and the other for the leader-unit-group to update the affected CTEK. The number of keys encrypted by the CKC during the rekeying is:

$$(1+1+2+3+\cdots+h_{\text{unit}})+n_{\text{leader\_unit}}=\frac{h_{\text{unit}}(h_{\text{unit}}+1)}{2}+n_{\text{leader\_units}}+1$$

where h_{unit} is the height of the key tree for the operation unit and $n_{\text{leader\_units}}$ is the number of leader units.

The group members in the directly-affected member unit receive two rekeying messages. One is the integrated message from the CKC updating the affected supporting keys within member unit, while the other is from the leader of the member unit updating the CTEK. The members outside the directly-affected member unit receive only one rekeying message from the unit leader to update the CTEK.

4.3.5.2 The Leaving Leader Unit

The leader leave process is more complex than the member departure process because three different scenarios (shown in Figure 4.7) can arise:

(i) a leadership candidate leaves the group;

(ii) a leader leaves the group and a leadership candidate is available to be the new leader; and

(iii) a leader leaves the group and no leadership candidate is available to be the new leader.

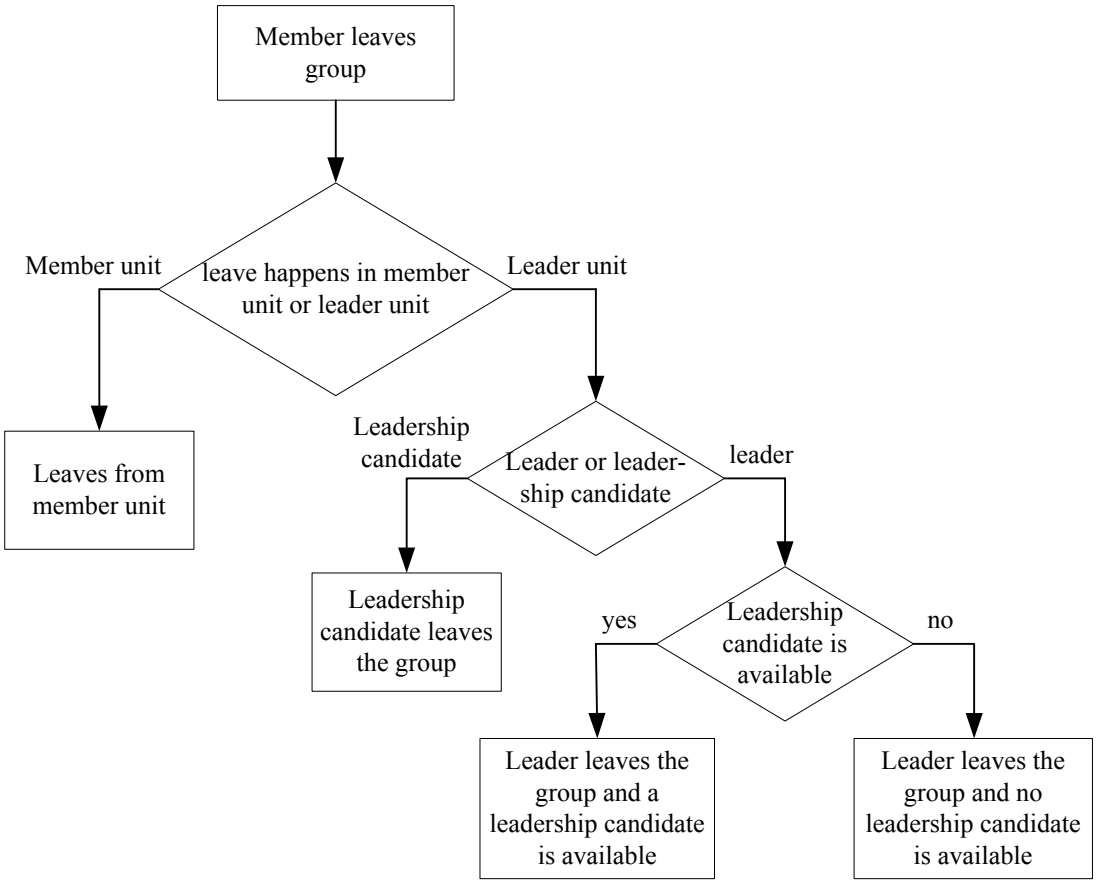


Figure 4.7 Three scenarios of leader's leave action

In the following sections, these three departure scenarios are discussed in depth.

4.3.5.3 A Leadership Candidate Leaves the Group

In this case, no member unit is affected by this departure because the leaving member has not yet been designated as a leader. The rekeying is restricted to the directly-affected leader unit that the leadership candidate leaves. Two steps are involved in the rekeying process.

- Step 1: Similar to the rekeying process of the leaving-from-member-unit, the CKC generates and multicasts an integrated rekeying message containing the

required rekeying information to the directly-affected leader unit.

- Step 2: For all the leader units, the CKC generates and multicasts a integrated rekeying message containing the copies of the newly-generated CTEK. Each copy is encrypted by the unit key of a leader unit. After receiving the new CTEK, the leaders distribute the new CTEK within their own member units.

We provide an example (Figure 4.8) to further illustrate the rekeying procedure when a leadership candidate leaves the group.

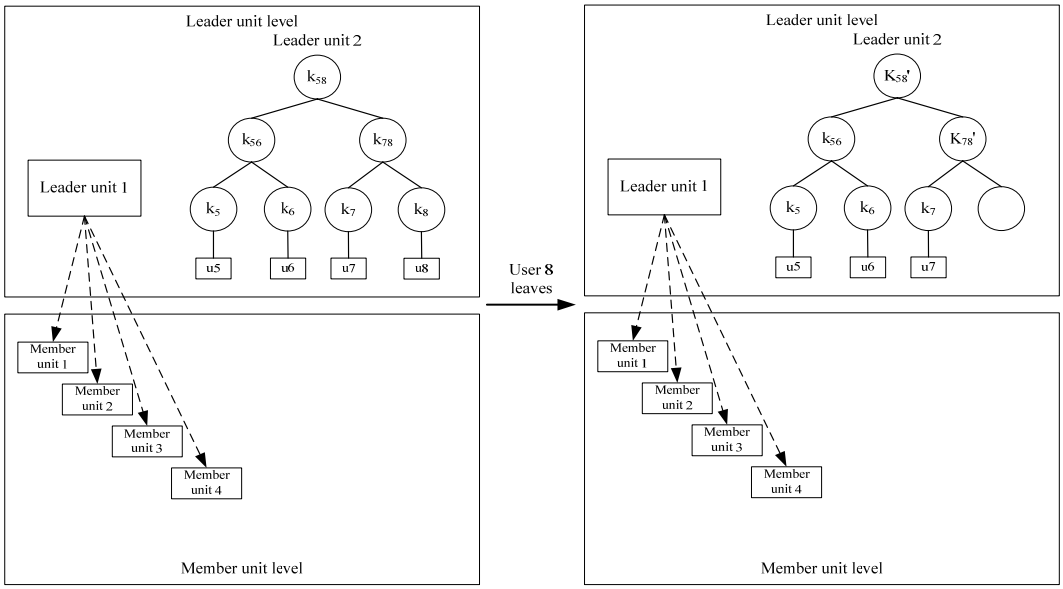


Figure 4.8 Scenario (i): A leadership candidate leaves the group

In Figure 4.8, the leadership candidate, user 8, leaves the group and the CKC generates the new keys $\{k_{\text{CTEK}}', k_{78}', k_{58}'\}$ to replace the affected current keys which are known by user 8. After key generation, the CKC invokes the rekeying procedure following a "bottom-to-top" method to update the keys for the remaining group members.

In step 1, the CKC generates a integrated message which contains all the required rekeying messages for the directly-affected operation unit, leader unit 2.

$$\text{CKC} \rightarrow \{\text{leader unit 2}\}: \{\text{for } \{\text{user 7}\}: \{k_{78}', k_{58}'\}k_7 \quad \text{for } \{\text{user 5,6}\}: \{k_{58}'\}k_{56}\}$$

When the members in leader unit 2 receive this message, they can obtain the useful rekeying message from the corresponding sections to update their keys. In this step, the CKC only sends a single rekeying message and the number of keys encrypted by the CKC is:

$$1 + 2 + \cdots + h_{\text{unit}} = \frac{h_{\text{unit}}(h_{\text{unit}} + 1)}{2}$$

where h_{unit} is the height of the key tree for the operation unit.

In step 2, after completing the rekeying in the directly-affected leader unit, the CKC generates another integrated rekeying message that contains the new CTEK for the leader units. The CKC then multicasts this integrated message within the leader units level via the leader-unit-group.

$$\text{CKC} \Rightarrow \{\text{leader-unit-group}\}: \{\{k_{\text{CTEK}}'\}k_{\text{leader\_unit\_1}}, \{k_{\text{CTEK}}'\}k_{\text{leader\_unit\_2}}\}$$

After receiving the new CTEK, the leaders distribute this new CTEK within their member units.

$$\text{user1} \Rightarrow \text{member unit1}: \{k_{\text{CTEK}}'\}k_{\text{member\_unit\_1}}$$
$$\text{user2} \Rightarrow \text{member unit2}: \{k_{\text{CTEK}}'\}k_{\text{member\_unit\_2}}$$
$$\text{user3} \Rightarrow \text{member unit3}: \{k_{\text{CTEK}}'\}k_{\text{member\_unit\_3}}$$
$$\text{user4} \Rightarrow \text{member unit4}: \{k_{\text{CTEK}}'\}k_{\text{member\_unit\_4}}$$

During this step, the CKC also only sends one single integrated rekeying message. The number of keys encrypted by the CKC is the number of leader units, $n_{\text{leader\_units}}$.

In this example, it can be observed that there are two integrated rekeying messages sent by the CKC during the whole rekeying process. The number of keys encrypted by the CKC is

$$(1+2+\cdots+h_{\text{unit}})+n_{\text{leader\_units}}=\frac{h_{\text{unit}}(h_{\text{unit}}+1)}{2}+n_{\text{leader\_units}},$$

where h_{unit} is the height of the key tree for the operation unit and $n_{\text{leader\_units}}$ is the number of leader units.

4.3.5.4 A Leader Leaves the Group and a Leadership Candidate Is Available to Be the New Leader

In the second scenario, when a leader leaves the group, the CKC can find an available leadership candidate to be the new leader of the directly-affected member unit. Rekeying can still be restricted to the directly-affected leader unit. The rekeying procedure of this scenario comprises of three steps, as follows.

- Step 1: the CKC generates an integrated message containing all rekeying messages to update the keys for the directly-affected leader unit the leader leaves.
- Step 2: the CKC chooses an available leadership candidate to be the new leader of the affected member unit. The CKC sends the new unit key and the new CTEK to the new chosen leader. After receiving these keys, the new leader distributes them within the newly-assigned member unit.
- Step 3: the CKC generates an integrated message containing the copies of the new CTEK. Each copy is encrypted by the unit key of a leader unit. After receiving the new CTEK, the leaders distribute the new CTEK within their

member units.

We present an example, shown in Figure 4.9, to explain this rekeying procedure further.

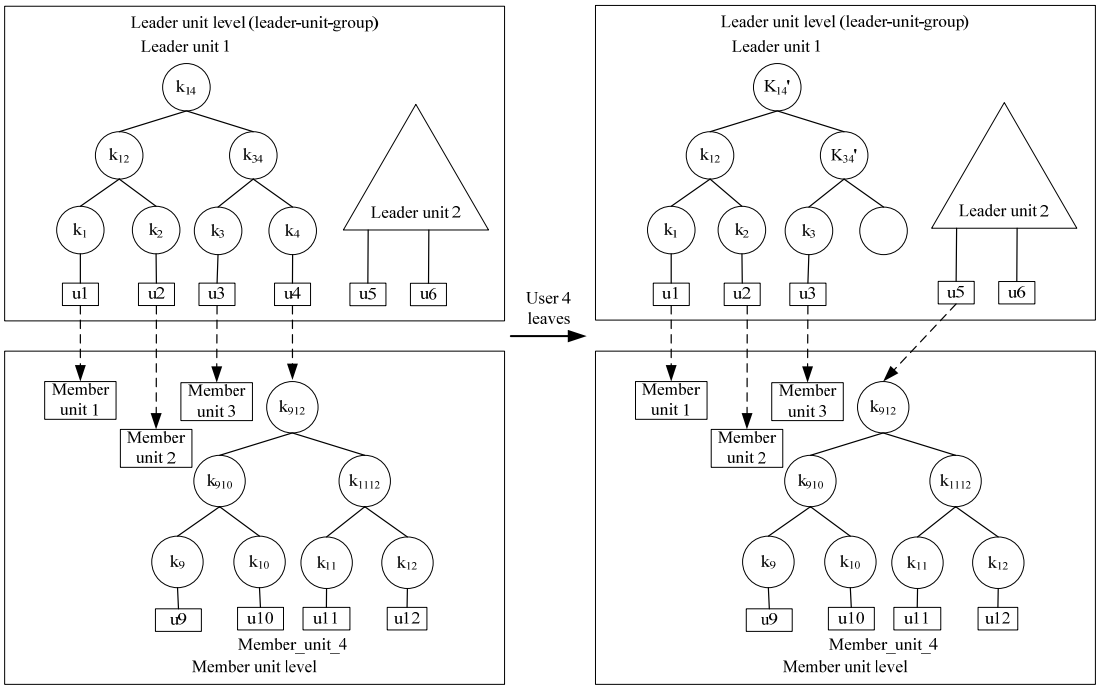


Figure 4.9 Scenario (ii): A leader leaves the group and a leadership candidate is available

In Figure 4.9, user 4 leaves the group and user 5 is selected as the new leader for the affected member unit 4. In order to ensure forward secrecy, the CKC generates the new keys $(k_{\text{CTEK}}', k_{34}', k_{14}', k_{912}')$ to replace the current ones that are known to the departing member, user 4. The key k_{912}' is the new unit key of directly-affected member unit 4 and it is known to the new leader, user 5.

After generating the new keys, the CKC invokes the rekeying procedure to update the keys of leader unit 1 that have been directly affected by the leave of user 4.

In step 1, the CKC generates an integrated message containing all the rekeying messages for leader unit 1 and sends it to the leader unit 1.

$$\text{CKC} \Rightarrow \{\text{leader unit 1}\}: \{\text{for } \{\text{user 3}\}: \{k_{34}', k_{14}',\} k_3 \text{ for } \{\text{user 1, 2}\}: \{k_{14}',\} k_{12}\}$$

After receiving this message, the members in leader unit 1 can obtain the latest keys from the corresponding sections. In this step, the CKC sends one rekeying message and the number of keys encrypted by the CKC is:

$$1 + 2 + \cdots + h_{\text{unit}} = \frac{h_{\text{unit}}(h_{\text{unit}} + 1)}{2}$$

where h_{unit} is the height of the key tree for the operation unit.

In step 2, the CKC designates user 5 to be the new leader of the affected member unit, member unit 4, and sends the new unit key and CTEK to user 5. After receiving these new keys, user 5 sends them to member unit 4.

$$\text{CKC} \rightarrow \{\text{user 5}\}: \{k_{912}', \{\{k_{912}', k_{\text{CTEK}}'\} k_{910}, \{k_{912}', k_{\text{CTEK}}'\} k_{1112}\}\} k_5$$

$$\text{user 5} \Rightarrow \{\text{member unit 4}\}: \{\{k_{912}', k_{\text{CTEK}}'\} k_{910}, \{k_{912}', k_{\text{CTEK}}'\} k_{1112}\}$$

During this step, the CKC also only sends one rekeying message. The number of keys encrypted by the CKC is five.

In the final step, the CKC generates an integrated message to update the CTEK for all members in the leader units.

$$\text{CKC} \Rightarrow \{\text{leader-unit-group}\}: \{\{k_{\text{CTEK}}'\} k_{\text{leader\_unit\_1}}, \{k_{\text{CTEK}}'\} k_{\text{leader\_unit\_2}}\}$$

After receiving the new CTEK, the leaders distribute the new CTEK within their own member units to update the key on behalf of the CKC.

$$\text{user } 1 \Rightarrow \{\text{member\_unit\_1}\}: \{k_{\text{CTEK}}'\}k_{\text{member\_unit\_1}}$$
$$\text{user } 2 \Rightarrow \{\text{member\_unit\_2}\}: \{k_{\text{CTEK}}'\}k_{\text{member\_unit\_2}}$$
$$\text{user } 3 \Rightarrow \{\text{member\_unit\_3}\}: \{k_{\text{CTEK}}'\}k_{\text{member\_unit\_3}}$$

In this step, CKC still sends one single rekeying message and the number of keys encrypted by the CKC equals the number of leader units, $n_{\text{leader\_units}}$.

In conclusion, during the rekeying procedure, the CKC needs to send three rekeying messages: one to update the keys in the directly-affected leader unit the leader leaves; one for the newly-chosen leader to update the unit key and CTEK for the directly-affected member unit due to the departure of the current leader; and finally, one to update the CTEK for all remaining members in the leader units. The total number of keys encrypted by the CKC during the rekeying is

$$(1+2+\cdots+h_{\text{unit}})+5+n_{\text{leader\_units}} = \frac{h_{\text{unit}}(h_{\text{unit}}+1)}{2}+5+n_{\text{leader\_units}},$$

where h_{unit} is the height of the key tree for the operation unit and $n_{\text{leader\_units}}$ is the number of leader units.

The members in the directly-affected leader unit receive two rekeying messages. One message updates the supporting keys within the unit while the other updates the CTEK. The newly-appointed leader receives two rekeying messages as well: one message containing the new unit key and the new CTEK for the directly-affected member unit and the other message rekeying the CTEK within the leader unit level. The members outside the directly-affected leader unit receive one single rekeying message to update the CTEK.

4.3.5.5 A Leader Leaves Group and No Leadership Candidate Is Available to Be the New Leader

In the final scenario, a leader leaves the group and the CKC cannot find an available leadership candidate to be the new leader of the directly-affected member unit whose leader leaves the group. Consequently, the CKC needs to upgrade the directly-affected member unit to a new leader unit. All the members in this newly-upgraded leader unit become leadership candidates. There are three steps in this rekeying process.

- Step 1: the CKC upgrades the directly-affected member unit to be a new leader unit and updates its unit key and the CTEK to enforce forward secrecy.
- Step 2: the CKC generates a integrated rekeying message to update the keys in the directly-affected leader unit from where the leader has departed.
- Step 3: the CKC generates another integrated rekeying message to update the CTEK within the leader unit level via the leader-unit-group. The leaders distribute the new CTEK to their own member units after receiving the new CTEK.

We provide an example, shown in Figure 4.10, to illustrate this rekeying procedure step by step. When user 4 in Figure 4.10 leaves the group, the CKC cannot find an available leadership candidate in the leader units to be the new leader of directly-affected member unit 4. Therefore, in step 1, the CKC upgrades member unit 4 to be a new leader unit. The CKC sends the new unit key to this new leader unit.

$$\text{CKC} \Rightarrow \{\text{member unit 4}\}:\{\{k_{912}'\}k_{910}, \{k_{912}'\}k_{1112}\}$$

In this step, the CKC sends one rekeying message and the number of keys encrypted by CKC is two.

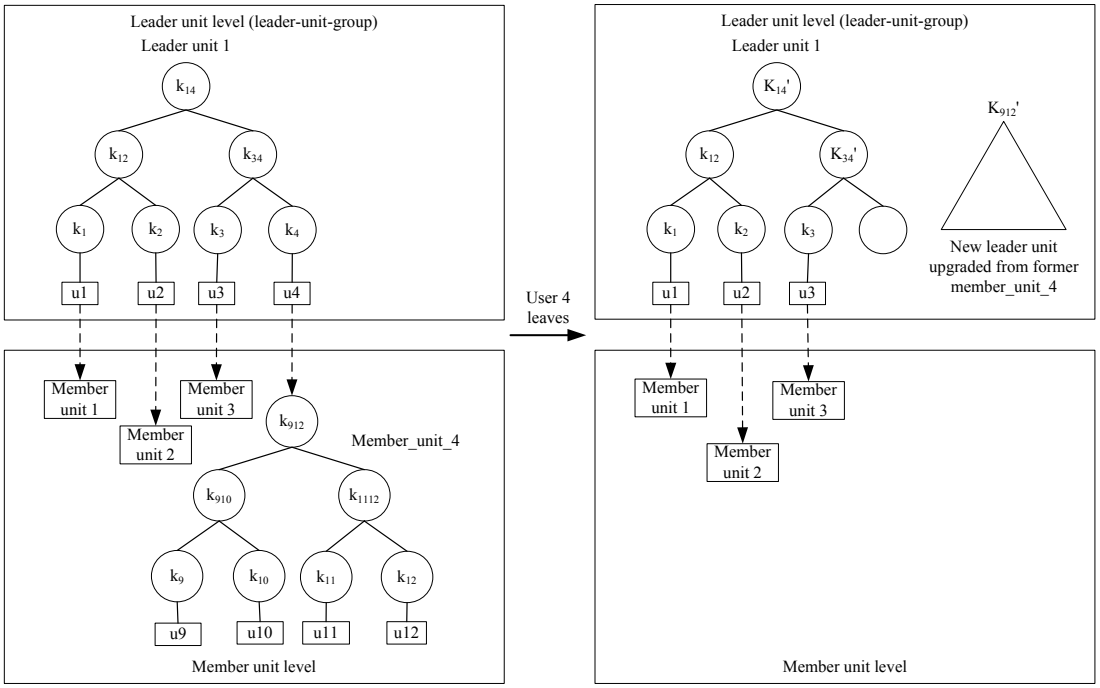


Figure 4.10 Scenario (iii): A leader leaves the group and no leadership candidate is available

In step 2, the CKC updates the supporting keys in the directly-affected leader unit, leader unit 1, by sending a integrated rekeying message.

$$\text{CKC} \rightarrow \{\text{leader unit 1}\}: \{\text{for } \{\text{user 3}\}: \{k_{34}', k_{14}',\} k_3 \quad \text{for } \{\text{user 1, 2}\}: \{k_{14}',\} k_{12}\}$$

The remaining leaders in leader unit 1 can update their keys after receiving this integrated rekeying message. In this step, the CKC also only sends one rekeying message and the number of keys encrypted by CKC is

$$1 + 2 + \cdots + h_{\text{unit}} = \frac{h_{\text{unit}}(h_{\text{unit}} + 1)}{2}$$

where h_{unit} is the height of the key tree for the operation unit.

In the final step, the CKC updates the CTEK for all leader units by sending another integrated rekeying message to the leader-unit-group. The leaders distribute the latest CTEK to their own member units after receiving the new key.

$$\text{CKC} \Rightarrow \{\text{leader-unit-group}\}: \{\{k_{\text{CTEK}}'\}k_{\text{leader\_unit\_1}}, \{k_{\text{CTEK}}'\}k_{912'}\}$$

$$\text{user 1} \Rightarrow \{\text{member\_unit\_1}\}: \{k_{\text{CTEK}}'\}k_{\text{member\_unit\_1}}$$

$$\text{user 2} \Rightarrow \{\text{member\_unit\_2}\}: \{k_{\text{CTEK}}'\}k_{\text{member\_unit\_2}}$$

$$\text{user 3} \Rightarrow \{\text{member\_unit\_3}\}: \{k_{\text{CTEK}}'\}k_{\text{member\_unit\_3}}$$

In step 3, the CKC sends one rekeying message and the number of keys encrypted by the CKC equals the number of leader units.

In summary, during the whole rekeying procedure, it can be observed that CKC sends 3 rekeying messages. Meanwhile, the number of keys encrypted by CKC is

$$(1+2+\cdots+h_{\text{unit}})+2+n_{\text{leader\_units}} = \frac{h_{\text{unit}}(h_{\text{unit}}+1)}{2}+2+n_{\text{leader\_units}}$$

where h_{unit} is the height of the key tree for the operation unit and $n_{\text{leader\_units}}$ is the number of leader units.

The members in the directly-affected leader unit receive two rekeying messages while the members in the newly-upgraded leader unit receive two rekeying messages as well. The members outside these two directly-affected units receive only one single rekeying message for updating the CTEK.

4.3.5.6 Summary of the Leave Operation

As a result of applying micro-key management within the operation unit, the main rekeying for the leave operation can be restricted to the directly-affected operation unit left by a member. For the CKC, because of the small size of the

operation unit, the size of each rekeying message is consequently small, and these rekeying messages can be placed in a single integrated message to utilize the capacity of multicast transmission to improve the communication efficiency. From the members' perspective, they only receive one or two targeted rekeying messages. This reduces the operational overhead for members. Table 4.2 summarizes the operational costs of rekeying during the leave operation for the CKC.

Table 4.2 The operational costs of the leave action for the CKC in HGKM

		The number of rekeying messages sent by CKC	The number of keys encrypted by CKC
Member unit leaving		2	$\frac{h_{\text{unit}}(h_{\text{unit}}+1)}{2}+n_{\text{leader\_units}}+1$
Leader unit leaving	Scenario (i)	2	$\frac{h_{\text{unit}}(h_{\text{unit}}+1)}{2}+n_{\text{leader\_units}}$
	Scenario (ii)	3	$\frac{h_{\text{unit}}(h_{\text{unit}}+1)}{2}+5+n_{\text{leader\_units}}$
	Scenario (iii)	3	$\frac{h_{\text{unit}}(h_{\text{unit}}+1)}{2}+2+n_{\text{leader\_units}}$

h_{unit} : the height of key tree for the operation unit

$n_{\text{leader\_units}}$: the number of leader units in the group

4.3.6 Key Management During Handoff

Mobility is the most important and unique feature in wireless networks. Mobility frees users from the restrictions of cables and provides users with network services anywhere and anytime. In the wireless environment, 'handoff' or 'handover' refers to the process by which a mobile terminal changes its network attachment

point. In the cellular wireless network, a mobile may be handed off from one wireless base station to another.

Handoff satisfies the convenience needs of wireless users, but it poses new challenges to group key management in relation to the distribution of keying materials when a member moves from one wireless cell to another. In order to address this problem, it is necessary to understand the current handoff scheme in the cellular wireless network (shown in Figure 4.11).

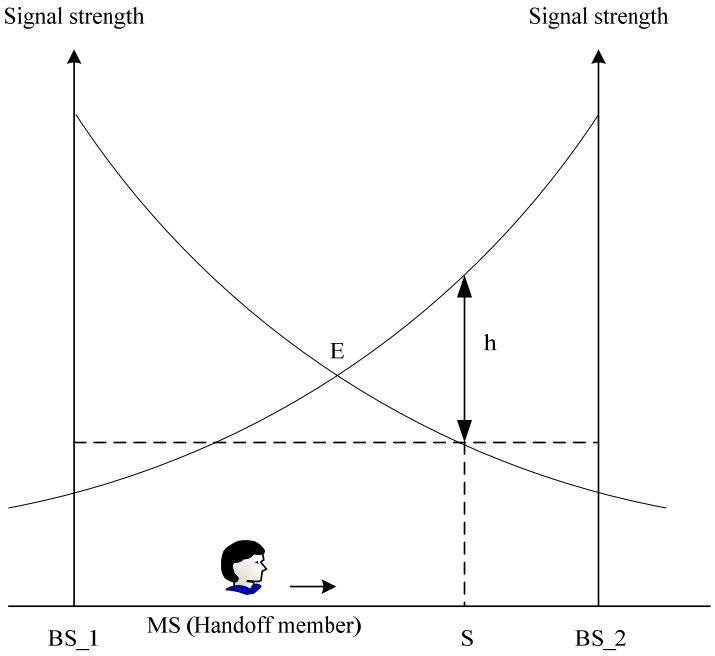


Figure 4.11 Handoff scheme in the cellular wireless network

In Figure 4.11, a mobile station (MS) is moving from one base station (BS_1) to another (BS_2). The mean signal strength of BS_1 decreases as the MS moves away from it. The mean signal strength of BS_2 increases as the MS approaches it. The parameter of hysteresis value is used to control the handoff operation. The hysteresis value measures the difference of the signal strength between the new BS and the

current BS. MS switches to the new BS when the signal is strong enough over a hysteresis margin (value h in Figure 4.11).

In the proposed group key management architecture for the cellular wireless network, each wireless cell is an independent administrative area and every group member registers to one and only one CKC for key management at any time. When a member moves from one cell to another, it detects a second signal from the destination BS. The current BS's signal decreases as the member moves away from it and the signal of the new BS increases. At the point where two signals are equal, point E in Figure 4.11, the member invokes the handoff join procedure by sending a handoff join request to the new CKC. After the new CKC receives this request, for the purpose of authentication, the new CKC sends a membership authentication request to the current CKC to identify the handoff member.

new CKC $\rightarrow$ current CKC: {authentication request for the handoff user}
current CKC $\rightarrow$ new CKC: {authentication reply}

If the authentication is successful, the new CKC searches for an available slot in its key management structure for this handoff user, and places the handoff member into its pre-handoff-user list. At this point, the handoff member is still registered to the current CKC. As the member keeps approaching the new BS, at point S in Figure 4.11, the difference in signal strength between the current BS and the destination BS exceeds the hysteresis value. At this point, the member switches to the destination BS and sends the handoff message to the new CKC. The new CKC registers the handoff member in its user list and sends its current CTEK to the user. The new CKC deletes the handoff user from the pre-handoff-user list, puts the handoff user into the

reserved slot in the key management structure and sends the keying materials to it. In addition, the new CKC sends a handoff confirmation message to the previous CKC to confirm the end of the handoff operation. After receiving this message, the previous CKC puts the member into a pro-handoff-user list and keeps the member in the list for a period of predefined time. If the member switches back before the time expires, there is no need for the CKC to invoke the leave rekeying. Otherwise, the previous CKC deletes the member from the pro-handoff-user list and marks the member's previous slot as being available for new incoming users. Figure 4.12 depicts the protocol of this handoff procedure.

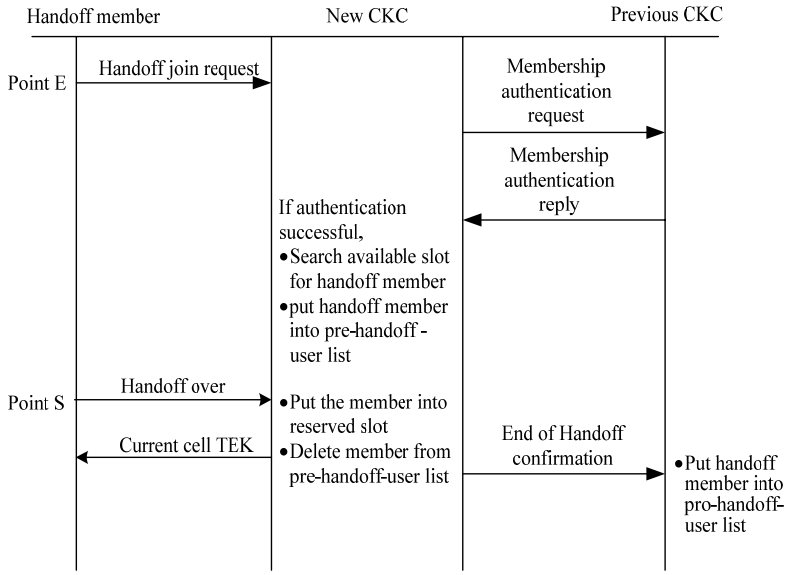


Figure 4.12 The key management protocol for the handoff procedure

The major advantage of this key management approach during the handoff is operational efficiency. Due to the cooperation of adjacent CKCs that allows members to have multiple sets of keys [DeCleene & Dondeti et al., 2001], the handoff key management procedure can be performed in parallel within the whole wireless

domain and operated without the interference of the group key controller (GKC).

4.3.7 Message Delivery

Message delivery is problematic in the wireless domain. Wireless communication is an unreliable transmission, because it is susceptible to the environment effects such as signal attenuation, reflection and shielding. Existing group key management approaches do not include reliable transmission mechanisms. While IP multicast transmission typically provides best-effort delivery from sender(s) to receivers, it cannot guarantee messages delivery. However, to ensure secure group communication, group members must receive up-to-date keying materials. Without these, members would not be able to participate in future group communication. A reliable delivery scheme must therefore be part of wireless group key management.

Unlike existing group key management approaches, HGKM has a built-in reliable message delivery scheme. In HGKM, retransmission agents are applied to guarantee message delivery. Each operation unit in HGKM can be treated as a self-assisted message delivery unit. At the leader unit level, all leader units are controlled by a CKC. Therefore, the CKC works as a retransmission agent to be responsible for the reliable delivery of keying materials to all members at the leader unit level. At the member unit level, the unit leader plays the same role as the CKC to ensure the reliable transmission of keying materials to group members.

We provide an example, shown in Figure 4.13, to further illustrate the operation of the reliable delivery scheme in HGKM. For user 4, in Figure 4.13, HGKM detects that user 4 lacks the latest keying materials as it cannot decrypt the group data.

Because user 4 is in leader unit 1, it sends a resending request to the CKC. After the CKC receives this request, the CKC sends user 4 a set of current keys which user 4 is entitled to know via unicast.

user 4 $\rightarrow$ CKC: {resending request for current key materials}

CKC $\rightarrow$ user 4: $\{k_{34}, k_{14}, k_{CTEK}\}k_4$

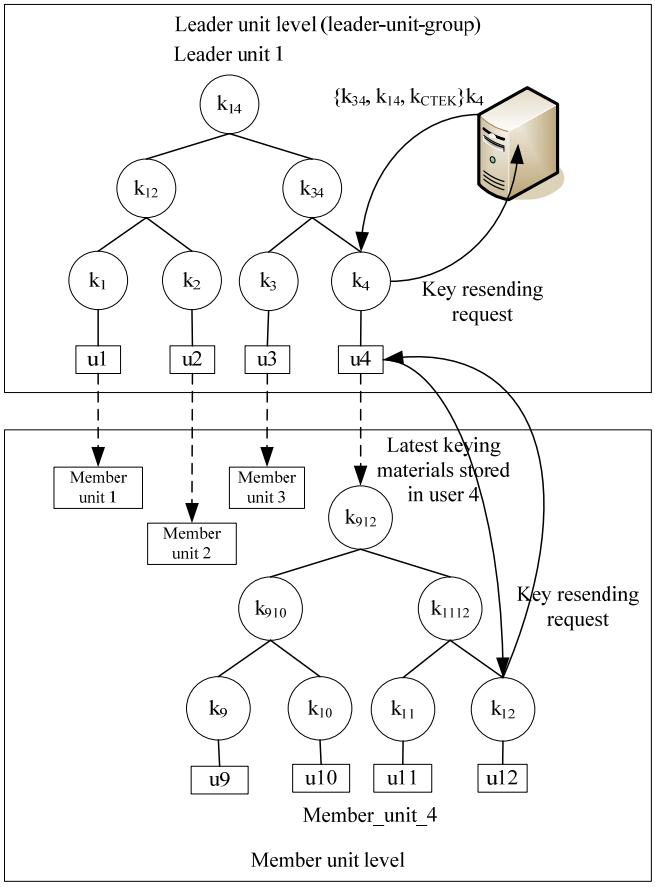


Figure 4.13 The reliable message delivery scheme in HGKM

Now, we consider the same scenario for user 12 in member unit 4 (shown in Figure 4.13). If user 12 realizes that it has missed the latest keying materials, it multicasts a resending request within the member unit where the unit leader listens for the resending requests. When user 4, the unit leader, receives this request, it sends

the latest keying materials that are stored on the side of user 4 to user 12 via unicast.

user 12 → user 4: {resending request for current keying materials}

user 4 → user 12: {the latest keying materials stored on the side of user 4}

In some cases, user 12 still cannot obtain the right keying materials from the unit leader by using this approach, since the unit leader only stores the latest round of keying materials and the member has missed more than two rounds of rekeying messages. In this case, the member directly contacts CKC to request a set of the current keys.

The reliable message delivery is the feature provided by HGKM in order to overcome the unreliable transmission of the wireless networks and IP multicasting. The advantages of this feature are operational efficiency and cost effectiveness.

- Operational efficiency

  The application of a two-tiered distributed retransmission approach ensures that most retransmissions are able to be performed within the member unit because the majority of group users are in the member units. Moreover, retransmission can be performed in parallel at the member unit level. This delivery approach improves system performance without substantially increasing the workload of the CKC.

- Cost-effectiveness

  The HGKM delivery approach is based on the operation units. It utilizes the logical key management structure in HGKM to form a built-in reliable transmission system. There is no need for an external message delivery system to guarantee message transmission.

4.3.8 Optimizing the Size of the Operation Unit

From the previous discussion, it can be observed that the operational costs of the join and leave procedure in HGKM are proportional to the height of the key tree for the operation unit and the number of leader units. Both of these are determined by the size of the operation unit. Therefore, the size of the operation unit is a critical parameter to determine the operational costs of key management in HGKM. Based on the investigation in sections 4.3.1 to 4.3.7, we realize that the size of the operation unit is affected by the following three factors: the number of leader units; the scope of reliable delivery; and the frequency of membership change.

- The number of leader units ($N_{\text{leader\_units}}$)

During the rekeying process, HGKM needs to generate a integrated rekeying message for all leader units. The size of this message is decided by the number of leader units. Figure 4.14 illustrates the number of leader units with different-sized operation units where we assume that the members in one leader unit operate as leadership candidates. In Figure 4.14, it can be observed that the smallest unit (in terms of size) has the largest number of leader units. This increases the number of the rekeying messages for the leader unit level. Therefore, in order to reduce the operational costs for the rekeying process (in terms of the number of rekeying messages), the number of operation units should be minimized.

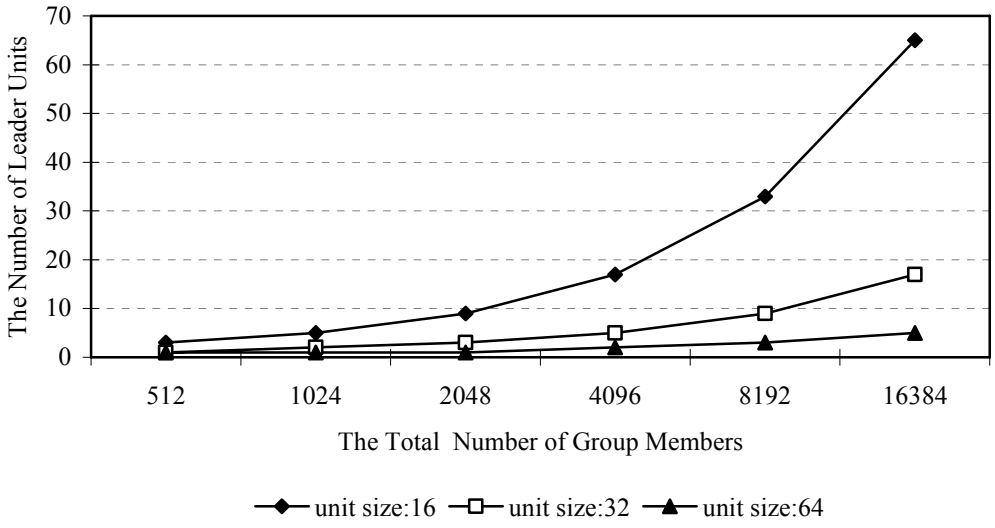


Figure 4.14 The number of leader units with different unit size

- The scope of reliable delivery ($S_{\text{reliable-delivery}}$)

In HGKM, each operation unit is also a reliable delivery group. If the size of the operation unit is too large, the flood of resending requests would overwhelm the processing capacity of the unit leader. Therefore, it is necessary to maintain the size of operation unit small.

- The frequency of membership change ($F_{\text{membership\_change}}$)

Members in an operation unit are organized in a hierarchical structure. This structure suffers operational inefficiency when applied to a highly dynamic group. It is necessary to keep the size of the operation unit small in order to reduce the operational costs and the impact of rekeying on the group members.

The size of the operation unit in HGKM, Size(unit), should therefore be determined by these three parameters:

$$\text{Size(unit)} = \{N_{\text{leader\_units}}, S_{\text{reliable-delivery}}, F_{\text{membership\_change}}\}$$

In this thesis, without loss of generality, the size of the operation unit is set at 32. We apply this size to analyze and evaluate operational costs of key management in HGKM in the following sections.

4.4 Performance Analysis

Operational efficiency is the highest priority in any wireless group key management scheme due to the resource limitations of both wireless networks and mobile devices. A wireless group key management approach cannot be recognized as efficient and practical if it cannot meet the requirements of operational efficiency. Therefore, in this section, we analyze and evaluate the operational costs of key management in HGKM to demonstrate that HGKM is an efficient and practical wireless group key management approach suitable for deployment in the cellular wireless network. In section 3.2.2, we discussed the fact that operational efficiency can be analyzed from three perspectives: communication cost, computation cost and key storage cost. In this section, we use these three parameters to perform analysis and evaluation. In order to evaluate the performance of HGKM, we set the centralized group key management approaches LKH and OFT as the benchmarks. The reasons for selecting these as benchmarks are: (i) LKH and OFT can be applied within the cell; (ii) LKH and OFT also apply a hierarchical key structure to manage keying materials which is similar to HGKM; and (iii) LKH and OFT are considered to be two of the most widely-used and efficient group key management approaches because the operational costs of these is $O(\log_d n)$ where d is the degree of the key

tree and n is the total number of group members.

4.4.1 Communication Cost

The cell key controller (CKC) is the main key management entity in HGKM. Consequently, the communication cost generally refers to the communication overhead of the CKC during the rekeying procedure caused by the join and leave procedures. The communication cost can be measured by the number of rekeying messages transmitted by the CKC during rekeying. Without loss of generality, we apply a binary tree to build the key management structure in HGKM, LKH and OFT, as a binary tree is easy to create, manage and maintain.

4.4.1.1 The Communication Cost of the Join Operation

In HGKM, there are two kinds of join actions: joining the member unit and joining the leader unit. Table 4.3 summarizes the communication cost of a single join action in HGKM, LKH and OFT, where the formulas are given in the sections 2.2 and 4.3.4.

Table 4.3 The communication cost of the join action for CKC

Group key management algorithm		Communication cost
HGKM	Join member unit	3
	Join Leader unit	3
LKH		$h+1$
OFT		$h+1$

h : the height of the key tree in LKH and OFT, which equals $\log_2 n$, where n is the group size.

From Table 4.3, it can be observed that the communication cost of the join action in LKH and OFT is proportional to the size of the whole group. Along with the increased group size, the communication cost of the join procedure in LKH and OFT also becomes larger.

In contrast, the communication cost of the join procedure for the CKC in HGKM is a constant value that is achieved by applying a integrated rekeying message to contain all the rekeying information for the affected operation unit. In HGKM, micro-key management is performed within a small fixed-sized operation unit. The number and the size of rekeying messages sent to the affected operation unit is therefore minimal. Because HGKM has the ability to place the rekeying messages into one single integrated message, it can utilize the capacity of multicast transmission to improve the throughput and efficiency of network transmission.

We apply the theory of expectation value [Grinstead & Laurie, 1991; Roberts, 1992] to calculate the average communication cost of a join action in HGKM.

$$\begin{aligned}\text{Cost}_{\text{communication}}(\text{join}) = {} & \text{Cost}_{\text{joining\_leader\_unit}} \times p_{\text{joining\_leader\_unit}} + \\ & \text{Cost}_{\text{joining\_memebr\_unit}} \times p_{\text{joining\_memebr\_unit}}\end{aligned}$$

where $\text{Cost}_{\text{joining\_leader\_unit}}$ and $\text{Cost}_{\text{joining\_memebr\_unit}}$ are the communication costs of joining leader unit and joining member unit respectively. $p_{\text{joining\_leader\_unit}}$ and $p_{\text{joining\_member\_unit}}$ presents the probability of joining leader unit and joining member unit correspondingly.

In HGKM, the probability of joining leader unit and joining member unit join are:

$$p_{\text{joining\_leader\_unit}} = \frac{n_{\text{members\_in\_leader\_units}}}{n_{\text{total\_group\_members}}}$$

$$p_{\text{joining\_member\_unit}} = \frac{n_{\text{members\_in\_member\_units}}}{n_{\text{total\_group\_members}}}$$

where $n_{\text{members\_in\_leader\_units}}$ is the number of members in the leader units, $n_{\text{members\_in\_member\_units}}$ is the number of members in the member units and $n_{\text{total\_group\_members}}$ is the total number of members in the whole group.

Therefore, the average communication cost of the join action in HGKM is

$$\begin{aligned}\text{Cost}_{\text{communication}}(\text{join}) &= 3 \times \frac{n_{\text{members\_in\_leader\_units}}}{n_{\text{total\_group\_members}}} + 3 \times \frac{n_{\text{members\_in\_member\_units}}}{n_{\text{total\_group\_members}}} \\ &= 3 \times \left(\frac{n_{\text{members\_in\_leader\_units}}}{n_{\text{total\_group\_members}}} + \frac{n_{\text{members\_in\_member\_units}}}{n_{\text{total\_group\_members}}}\right) \\ &= 3\end{aligned}$$

Based on this calculation, it can be observed that the average communication cost of the join action for HGKM is a constant value, equaling 3.

The comparison of the communication cost of the join action in HGKM, LKH and OFT is shown in Figure 4.15.

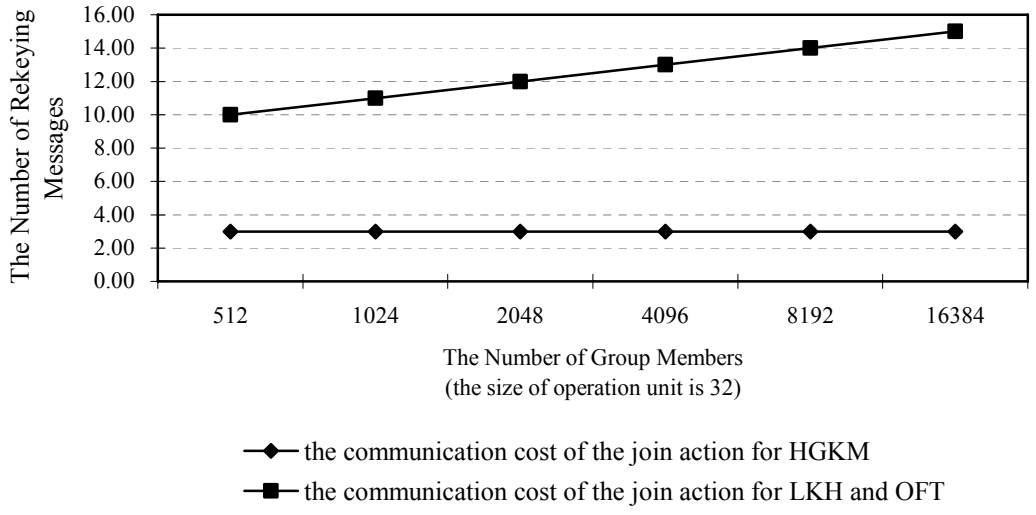


Figure 4.15 The comparison of communication cost of the join action

From Figure 4.15, we can observe the following three features:

- the communication cost of the join action in HGKM is a constant value independent of the size of the group ;
- the communication cost of the join action in LKH and OFT is three to five times higher than that of HGKM. Moreover, the communication cost of the join action in LKH and OFT increases with the growth of the number of group members; and
- the gap in the communication cost between HGKM and LKH and OFT becomes increasingly wider as the group size increases.

These three features ensure that HGKM can achieve much better communication efficiency in key management during the join procedure than the LKH and OFT approaches.

4.4.1.2 The Communication Cost of the Leave Operation

In HGKM, there are four scenarios to be considered when a member or leader leaves the group:

(i) a member leaves a member unit;

(ii) a leadership candidate leaves the group;

(iii) a leader leaves the group and a leader candidate is available to be the new leader of the affected member unit; and

(iv) a leader leaves the group and no leader candidate is available to be the new leader.

Table 4.4 tabulates the communication cost of the leave operation for the CKC, where the values are from the sections 2.2 and 4.3.5.

Table 4.4 The communication cost of the leave action for HGKM, LKH and OFT

Group key management algorithm		Communication cost
HGKM	Scenario (i): Member leaves a member unit	2
	Scenario (ii): a Leadership candidate leaves leader unit	2
	Scenario (iii): a Leader leaves the group and a leader candidate available to be the new leader	3
	Scenario (iv): a Leader leaves the group and no leader candidate is available	3
LKH		h
OFT		h

h : the height of the group key tree, which equals $\log_2 n$, where n is the size of whole group.

In HGKM, scenario (iii) and (iv) are mutually exclusive and only one scenario occurs when a leader leaves the group. Therefore, we classify the leave operation into two cases comprising the above four scenarios as follows:

- Case I: scenario (i), scenario (ii) and scenario (iii); and
- Case II: scenario (i) and scenario (iv).

Based on each case, we can apply the theory of expectation value to compute the average communication cost of the leave operation for HGKM. Before we perform this calculation, we assume that the total number of group members is $n_{\text{total\_group\_members}}$, the number of leaders in the leader unit is n_{leaders} , the number of leadership candidates is $n_{\text{leadership\_candidates}}$ and the total number of members in all

member units is $n_{\text{members\_in\_member\_units}}$. Moreover, the total number of group members equals the summation of the other three groups:

$$n_{\text{total\_group\_members}} = n_{\text{leaders}} + n_{\text{leadership\_candidates}} + n_{\text{members\_in\_member\_units}}$$

(i) The average communication cost of the leave action for Case I:

In Case I, the probability of each scenario is:

- Scenario (i) (a member leaves member unit):

$$p_1(\text{Case I}) = \frac{n_{\text{members\_in\_member\_units}}}{n_{\text{total\_group\_members}}}$$

- Scenario (ii) (a leadership candidate leaves the group):

$$p_2(\text{Case I}) = \frac{n_{\text{leadership\_candidates}}}{n_{\text{total\_group\_members}}}$$

- Scenarios (iii): (a leader leaves the group and a leadership candidate is available to be the new leader):

$$p_3(\text{Case I}) = 1 - p_1(\text{Case I}) - p_2(\text{Case I}) = \frac{n_{\text{leaders}}}{n_{\text{total\_group\_members}}}$$

Thus, the average communication cost of the leave action for Case I is

$$\begin{aligned}\text{Cost}_{\text{communication\_leaving}}(\text{Case I}) &= 2 \times p_1(\text{Case I}) + 2 \times p_2(\text{Case I}) + 3 \times p_3(\text{Case I}) \\ &= 2 \times (1 - p_2(\text{Case I}) - p_3(\text{Case I})) + 2 \times p_2(\text{Case I}) + 3 \times p_3(\text{Case I}) \\ &= 2 + p_3(\text{Case I}) \\ &= 2 + \frac{n_{\text{leaders}}}{n_{\text{total\_group\_members}}}\end{aligned}$$

(ii) The average communication cost of the leave action for Case II

In Case II, there is no leadership candidate in the group. When a leader leaves the group, the CKC needs to upgrade the affected member unit into a new leader unit. Therefore, the probability of each scenario is:

- Scenario (i) (a member leaves member unit):

$$p_1(\text{Case II}) = \frac{n_{\text{members\_in\_member\_units}}}{n_{\text{total\_group\_members}}}$$

- Scenario (iv) (a Leader leaves the group and no leadership candidate is available):

$$p_4(\text{Case II}) = 1 - p_1(\text{Case II}) = \frac{n_{\text{leaders}}}{n_{\text{total\_group\_members}}}$$

The average communication cost of the leave action for case II is

$$\begin{aligned}\text{Cost}_{\text{communication\_leaving}}(\text{Case II}) &= 2 \times p_1(\text{Case II}) + 3 \times p_4(\text{Case II}) \\ &= 2 \times (1 - p_1(\text{Case II})) + 3 \times p_4(\text{Case II}) \\ &= 2 + p_4(\text{Case II}) \\ &= 2 + \frac{n_{\text{leaders}}}{n_{\text{total\_group\_members}}}\end{aligned}$$

From the above calculation, it can be observed that the average communication costs of the leave action for Cases I and II is the same. Both of them equal $2 + \frac{n_{\text{leaders}}}{n_{\text{total\_group\_members}}}$.

Figure 4.16 illustrates an example of the average communication cost of the leave procedure for both cases with the various group sizes. We assume that the size of the operation unit in HGKM is 32 and members in one leader unit operate as leadership candidates.

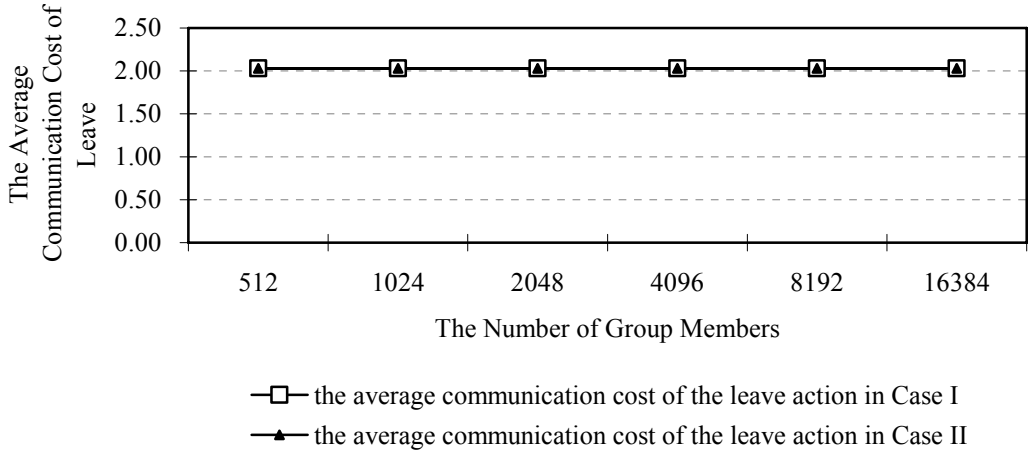


Figure 4.16 The communication cost of the leave action for Cases I and II

From Figure 4.16, it can be observed that the average communication cost of the leave action for both cases is the same, and is a constant value independent of the change in group size. This is because the average communication cost of a leave action for both cases can be simplified to $2+\frac{1}{s_{\text{operation\_unit}}}$, where $s_{\text{operation\_unit}}$ is the size of the operation unit. Based on the above calculation, the average communication cost of the leave action for both cases equals $2+\frac{n_{\text{leaders}}}{n_{\text{total\_group\_members}}}$. The number of leaders, n_{leaders}, is determined by the group size ($n_{\text{total\_group\_members}}$) and operation unit size ($s_{\text{operation\_unit}}$)

$$n_{\text{leaders}} = \frac{n_{\text{total\_group\_members}}}{s_{\text{operation\_unit}}}.$$

Thus, we can simplify the average communication cost of the leave action as:

$$\text{Cost}_{\text{communication}}(\text{leaving}) = 2+\frac{n_{\text{leaders}}}{n_{\text{total\_group\_members}}} = 2+\frac{1}{s_{\text{operation\_unit}}}.$$

When the size of operation unit is determined (as in Figure 4.16), both cases have a same constant communication cost.

A comparison of the communication cost of the leave action for HGKM, LKH and OFT is illustrated in Figure 4.17.

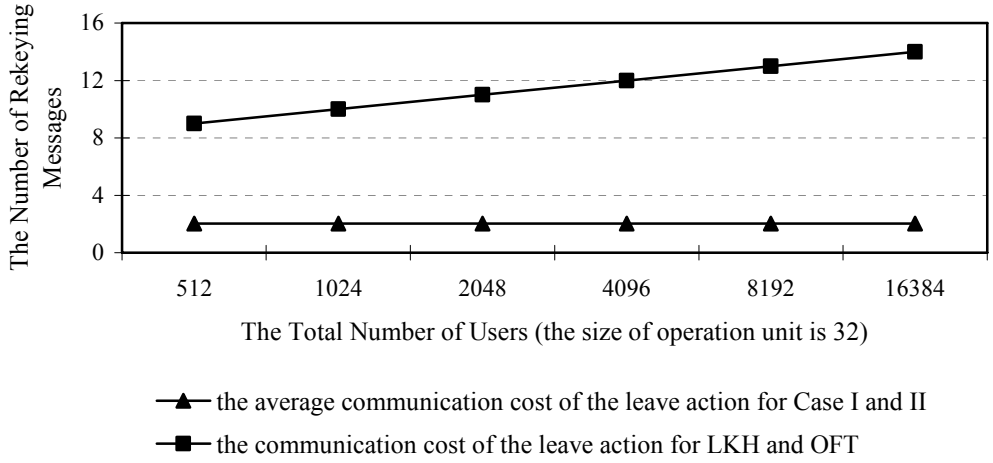


Figure 4.17 The communication cost of the leave action for HGKM, LKH and OFT

From Figure 4.17, it can be observed that the communication cost of the leave action is similar to that of the join operation with the following three features.

- The communication cost of the leave action for both cases in HGKM is the same, and is independent of the variation of group size. The communication cost is a constant value when the size of operation unit can be determined.
- In contrast, the communications cost of the leave action for LKH and OFT is the same and up to four to seven times higher than that of HGKM. Furthermore, the communications cost for LKH and OFT increases logarithmically as the group grows.

- With the increase in group size, the cost gap between HGKM, and LKH and becomes increasingly wider.

Based on the above analysis and evaluation, HGKM performs the leave operation more efficiently than LKH and OFT.

4.4.1.3 Summary

Based on the above analysis and evaluation, the communication cost of the join and leave operations for HGKM are constant values that are achieved by applying micro-key management within the operation unit. In HGKM, group members are divided into a number of small operation units and rekeying can be performed within the small scope of the operation unit. Due to the performance of micro-key management, the communication cost of the join and leave procedures for HGKM are reduced compared to those for LKH and OFT. Moreover, due to the small number of rekeying messages during key updating, all related rekeying messages can be placed in a single integrated message for transmission. This can utilize the capacity of multicasting and further reduce the communication cost and improve transmission efficiency. In conclusion, HGKM is able to achieve greater operational efficiency in communication than LKH and OFT during the rekeying of the join and leave operations.

Table 4.5 summarizes the average communication cost of the join and leave operation for HGKM.

Table 4.5 The average communication cost of the join and leave actions for HGKM

Wireless group key management	The average communication cost of the join action	The average communication cost of the leave action
HGKM	3	$2+\frac{1}{s_{\text{operation\_unit}}}$

$s_{\text{operation\_unit}}$: the size of the operation unit in HGKM

4.4.2 Computation Cost

Computation cost is another important parameter that can be used to evaluate the operational efficiency of group key management. The task of encryption and decryption is the heaviest work done by the CKC and group members during rekeying, making it an appropriate criterion for assessment. As discussed in section 3.2.2, for the CKC, this parameter can be measured by the number of keys encrypted during the rekeying procedure. For group members, this parameter can be approximately assessed by ascertaining the number of rekeying messages received by members. During rekeying, every rekeying message is multicasted to all group members to improve communication efficiency. Each member needs to process every single received message to determine if it is the intended recipient. The computation cost for member is therefore directly related to the number of received messages. In the next two sections, we analyze the computation costs of the join and leave operation respectively. We set the costs of LKH and OFT as the benchmarks for the purpose of comparison.

4.4.2.1 The Computation Cost of the Join Operation

(i) the computation cost of the join operation for the CKC

Table 4.6 summarizes the computation cost of a single join operation in HGKM, LKH and OFT, where the formulas are from the sections 2.2, 4.3.4 and 4.3.5.

Table 4.6 The computation cost of the join operation for CKC

Group key management	Computation cost of the join action for CKC	
	Leader unit	Member unit
HGKM	$\frac{(h_{\text{unit}}+1)(h_{\text{unit}}+2)}{2}+1$	$\frac{(h_{\text{unit}}+1)(h_{\text{unit}}+2)}{2}+2$
LKH	$\frac{(h+1)(h+2)}{2}-1$	
OFT	$2h$	

h_{unit} : the height of the key tree for operation unit in HGKM

h : the height of the group key tree in LKH and OFT

In Table 4.6, it can be observed that the computation cost of the join action in HGKM is proportional to the power of the height of the key tree for the operation unit, $O({h_{\text{unit}}}^2)$. This cost is independent of the size of the group members. Once the size of the operation unit is determined, the computation cost of the join operation for HGKM becomes a constant value. In contrast, in terms of LKH, the computation cost of the join operation is proportional to the power of the height of the group key tree, $O(h^2)$, while the computation cost of the join action for OFT is in ratio to the height of the group key tree, $O(h)$, which is due to the dependent key generation by passing the old key through a one-way function to obtain a new key.

In section 4.3.4, we introduced two join scenarios in HGKM: joining leader unit and joining member unit. Based on the theory of expectation value, for a single join action, the average computation cost is:

$$Cost_{\text{computation}}(\text{join}) = (\frac{(h_{\text{unit}}+1)(h_{\text{unit}}+2)}{2}+1)\times p_{\text{leader\_unit}}(\text{join}) + (\frac{(h_{\text{unit}}+1)(h_{\text{unit}}+2)}{2}+2)\times p_{\text{member\_unit}}(\text{join})$$

where $p_{\text{leader\_unit}}(\text{join})$ and $p_{\text{member\_unit}}(\text{join})$ are the probability of joining leader unit and probability of joining member unit respectively. In HGKM, the probability of joining leader unit and joining member unit are:

$$p_{\text{leader\_unit}}(\text{join}) = \frac{n_{\text{members\_in\_leader\_units}}}{n_{\text{total\_group\_members}}}$$
$$p_{\text{member\_unit}}(\text{join}) = 1 - p_{\text{leader\_unit}}(\text{join})$$

where $n_{\text{members\_in\_leader\_units}}$ is the number of members in the leader units and $n_{\text{total\_group\_members}}$ is the total number of member in the group. Therefore, the average computation cost of the join action for HGKM can be simplified as follows:

$$\begin{aligned}\text{Cost}_{\text{computation}}(\text{join}) &= (\frac{(h_{\text{unit}}+1)(h_{\text{unit}}+2)}{2}+1)\times p_{\text{leader\_unit}}(\text{join}) + \\ &\quad (\frac{(h_{\text{unit}}+1)(h_{\text{unit}}+2)}{2}+2)\times p_{\text{member\_unit}}(\text{join}) \\ &= \frac{(h_{\text{unit}}+1)(h_{\text{unit}}+2)}{2}+2-\frac{n_{\text{members\_in\_leader\_units}}}{n_{\text{total\_group\_members}}}\end{aligned}$$

The comparison of computation cost of join for the CKC in HGKM, LKH and OFT is shown in Figure 4.18. In this example, we assume that the size of the operation unit is 32.

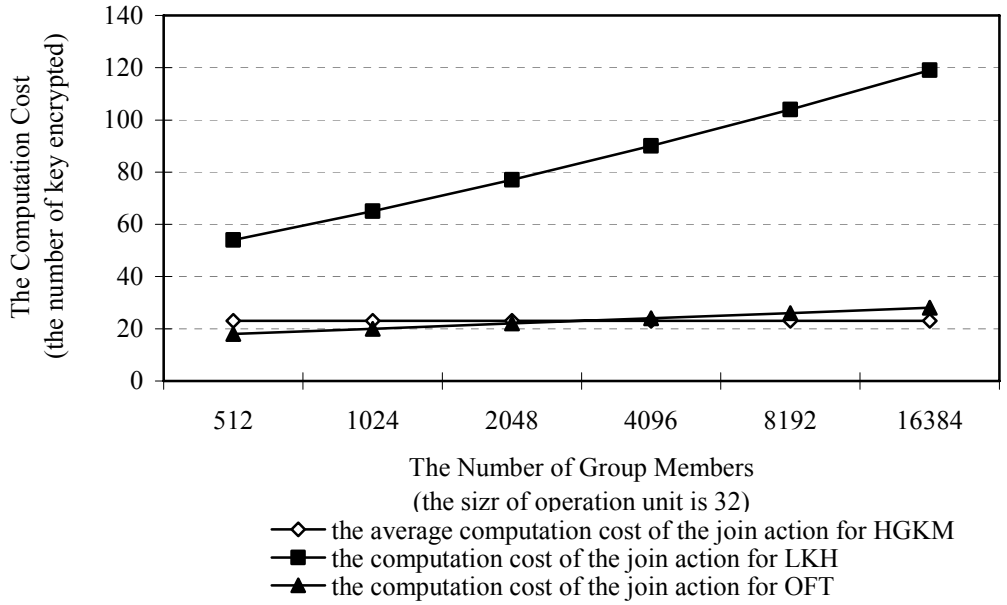


Figure 4.18 The computation cost of the join action for the CKC

In Figure 4.18, three features of the computation cost of the join action for the CKC can be observed.

- Due to the application of dependent key generation, the computation cost of the join action for the CKC in OFT is low and proportional to the height of the group key tree, $O(h)$.
- HKGM can achieve the same level efficiency as that of OFT because the rekeying process is confined within the small sized operation unit. The computation cost of the join action for HGKM is insensitive to increasing group size. Despite the growing in the number of group members, the computation cost of the join operation for HGKM is little changed. The reason for this is the application of micro-key management within the scope of the operation unit;

- LKH has the highest computation cost of the three approaches, with a computation cost of the join action two to six times higher than that of HGKM. Moreover, the computation cost of the join action for LKH increases logarithmically as the group grows. The gap in the computation cost between LKH and HGKM consequently becomes increasingly wider.

In conclusion, HGKM has an advantage over LKH in the computation cost of the join action, with the cost being similar to that of OFT. Moreover, the advantage increases with the growth of the group size.

(ii) The computation cost of the join operation for members

For group members, the computation cost can be measured by the number of received rekeying messages. As discussed in section 2.2, in the centralized group key management approaches, each rekeying message is multicasted to the whole group. However, it is only useful to a set of members and not necessarily the whole group. Nonetheless, a member needs to process all received rekeying messages to find the single message for which it is targeted. For a large and highly dynamic group, frequent rekeying may overwhelm the processing capacity of lightweight mobile devices. Overall, this rekeying approach represents an inefficient use of resources.

In order to address this operational efficiency issue, HGKM applies small operation units to perform micro-key management. Most of the rekeying operations can be confined within the directly-affected operation unit where the join and leave operations take place. Due to the small size of the operation unit, HGKM can reduce the number of rekeying messages transmitted during rekeying. Furthermore, due to the small number and size of these rekeying messages, they can be placed into a

integrated message for more efficient transmission. As a result of this transmission, members in the directly-affected operation unit only need to process this integrated message to find the relevant information. For the purpose of comparison, in HGKM, we assume that the computation cost of the join action for the members in the directly-affected operation unit is the number of rekeying messages contained in the integrated package. Table 4.7 summarizes the computation cost of the join action for members in HGKM, LKH and OFT, where the formula is from the previous sections 2.2 and 4.3.4.

Table 4.7 The computation cost of the join operation for group members

Group key management algorithm	Joining leader unit		Joining member unit	
	Members in the directly-affected operation unit	Members outside the directly-affected operation unit	Members in the directly-affected operation unit	Members outside the directly-affected operation unit
HGKM	$h_{\text{unit}}+1$	1	$h_{\text{unit}}+1$	1
LKH	h			
OFT	h			

h_{unit} : the height of the key tree for operation unit in HGKM

h : the height of the group key tree, which equals $\log_2 n$, n is the total number of group members.

From Table 4.7, it can be observed that the computation cost of join for the members in the directly-affected operation unit is proportional to the height of the key tree for the operation unit in both join scenarios. In contrast, the computation cost of the join action for LKH and OFT equals the height of the group key tree. Compared to the size of key tree for LKH and OFT, HGKM has small operation units. HGKM consequently has better computation efficiency than LKH and OFT. In

addition, the most important improvement in the computation cost for HGKM is that the computation cost of the join action for group members outside the directly-affected operation unit is just one. This achievement minimizes the impact of rekeying on the remaining members and drastically improves the computation efficiency for members.

Figure 4.19 illustrates the computation cost of join on the members' side for HGKM, LKH and OFT. In this example, we suppose that the size of the operation unit is 32.

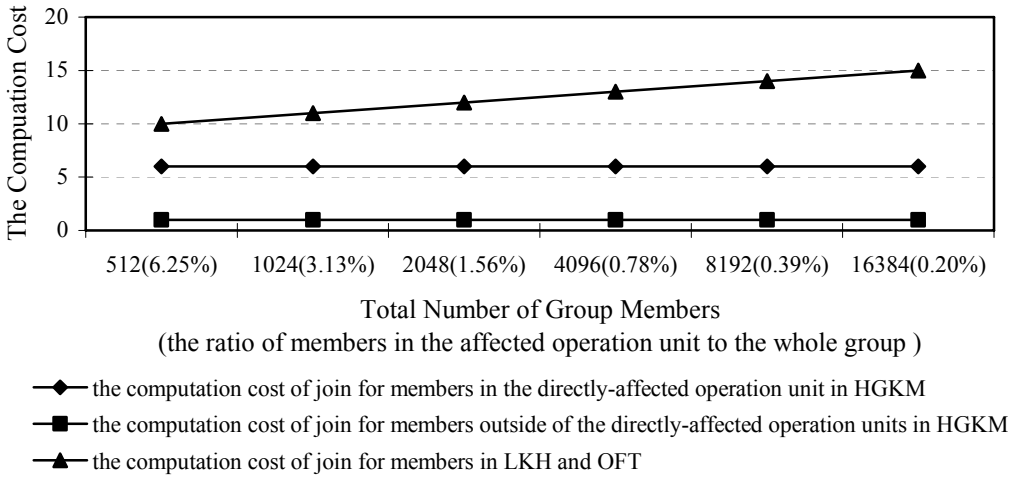


Figure 4.19 The computation cost of the join action for members

From Figure 4.19, it can be observed that the computation cost of the join action for the members in the directly-affected operation unit in HGKM is a constant value and independent of the group size. The reason for this is that the height of the key tree for the operation unit is determined by the size of operation unit. In this example, the size of operation unit is fixed. Due to the small size of operation unit, the computation cost of the join action for the members within the directly-affected

operation unit is reduced compared to that of LKH and OFT. Moreover, the members in the directly-affected operation unit are a very small portion of the whole group (the percentage is shown in Figure 4.19). This demonstrates that the micro-key management can reduce the impact of key updating to the whole group because the rekeying is confined within the scope of the operation unit. For the majority of members in HGKM who are outside the directly affected operation unit, the computation cost of join is only one. This reduces the computation burden for members, especially for the members in a large and highly dynamic group. This feature makes HGKM particularly suitable for wireless networks.

For LKH and OFT, the computation cost of the join action is one to two times that of the members in the directly-affected operation unit, and more than ten times that of the members outside the directly-affected operation unit. Furthermore, the computation cost of the join action for LKH and OFT increases logarithmically in relation to the growth of the group size. The gap in the computation cost of join between HGKM, and LKH and OFT becomes steadily wider as the group size increases.

To sum up, for members, based on the above analysis and evaluation, HGKM can achieve a better performance in the computation cost of the join action than LKH and OFT.

4.4.2.2 The computation cost of the leave operation

(i) Computation cost of the leave operation for the CKC

As discussed in section 4.2.4, for the CKC in HGKM, four leave scenarios need

to be considered when analyzing and evaluating the computation cost of the leave action. They are:

- Scenario (i): a member leaves from a member unit
- Scenario (ii): a leader candidate leaves the group
- Scenario (iii): a leader leaves the group and a leadership candidate is available to be the new leader
- Scenario (iv): a leader leaves the group and no leadership candidate is available to be the new leader.

Table 4.8 tabulates the computation costs of the leave action for the CKC.

Table 4.8 The computation cost of the leave action for the CKC

Group key management approaches		Computation cost
HGKM	Scenario (i): Member leaves member unit	$\frac{h_{\text{unit}}(h_{\text{unit}}+1)}{2}+n_{\text{leader\_units}}+1$
	Scenario (ii): A leader candidate leaves the group	$\frac{h_{\text{unit}}(h_{\text{unit}}+1)}{2}+n_{\text{leader\_units}}$
	Scenario (iii): A leader leaves the group and a leadership candidate is available to be the new leader	$\frac{h_{\text{unit}}(h_{\text{unit}}+1)}{2}+5+n_{\text{leader\_units}}$
	Scenario (iv): A leader leaves the group and no leadership candidate is available	$\frac{h_{\text{unit}}(h_{\text{unit}}+1)}{2}+2+n_{\text{leader\_units}}$
LKH		$\frac{h(h+1)}{2}$
OFT		h

h_{unit} : the height of the operation unit in HGKM

h : the height of the group key tree in LKH and OFT

$n_{\text{leader\_units}}$: the number of leader units in HGKM

In Table 4.8, we can observe that, in HGKM, the computation costs of the leave action for all four scenarios rely on the height of the key tree for the operation unit and the number of leader units. In contrast, the computation cost of the leave action for LKH depends on the power of height of the group key tree, $O(h^2)$, where h is the height of the group key tree. The computation cost of leave for the CKC in OFT is only proportional to the height of the group key tree, $O(h)$. This is because the new key is generated locally by applying a one-way function on the members' side. This reduces the computation workload of the CKC in OFT.

As discussed in section 4.4.1.2, we categorize these four scenarios into two cases, Case I and Case II. Case I comprises Scenarios i, ii, and iii and Case II comprises Scenarios i and iv. We assume that the total number of group members is $n_{\text{total\_group\_members}}$, the number of leaders in all leader units is n_{leaders}, the number of leadership candidates is $n_{\text{leadership\_candidates}}$ and the total number of the members in all member units is $n_{\text{members\_in\_member\_units}}$. The total number of group members equals the summation of the other three groups.

$$n_{\text{total\_group\_members}} = n_{\text{leaders}} + n_{\text{leadership\_candidates}} + n_{\text{members\_in\_member\_units}}$$

We apply the theory of expectation value to calculate the average computation cost for the CKC as follows:

(i) Case I

In Case I, the probability of each scenario is:

- Scenario (i): a member leaves a member unit

$$p_1(\text{Case I}) = \frac{n_{\text{members\_in\_member\_units}}}{n_{\text{total\_group\_members}}};$$

- Scenario (ii): a leader candidate leaves the group

$$p_2(\text{Case I}) = \frac{n_{\text{leadership\_candidates}}}{n_{\text{total\_group\_members}}};$$

- Scenario (iii): a leader leaves the group and a leader candidate available

$$p_3(\text{Case I}) = 1 - p_1(\text{Case I}) - p_2(\text{Case I}).$$

The average communication cost of the leave action for Case I in HGKM is

$$\begin{aligned}\text{Cost}_{\text{comuputation\_leave}}(\text{Case I}) &= \text{Cost}_{\text{scenario\_1}} \times p_1(\text{Case I}) + \text{Cost}_{\text{scenario\_2}} \times p_2(\text{Case I}) + \\ &\quad \text{Cost}_{\text{scenario\_3}} \times p_3(\text{Case I}) \\ &= \text{Cost}_{\text{scenario\_1}} \times p_1(\text{Case I}) + \text{Cost}_{\text{scenario\_2}} \times p_2(\text{Case I}) + \\ &\quad \text{Cost}_{\text{scenario\_3}} \times (1 - p_1(\text{Case I}) - p_2(\text{Case I})) \\ &= \frac{h_{\text{unit}}(h_{\text{unit}}+1)}{2} + 5 + n_{\text{leader\_units}} - 4 \times p_1(\text{Case I}) - \\ &\quad 5 \times p_2(\text{Case I})\end{aligned} \quad ($$

ii) Case II

In case II, the probability of each scenario is:

- Scenario (i): a member leaves a member unit

$$p_1(\text{Case II}) = \frac{n_{\text{members\_in\_member\_units}}}{n_{\text{total\_group\_members}}};$$

- Scenario (iv): a leader leaves the group and no leader candidate available:

$$p_4(\text{Case II}) = 1 - p_1(\text{Case II}).$$

The average communication cost of the leave action for Case II in HGKM is

$$\text{Cost}_{\text{comuputation\_leaving}}(\text{case II}) = \text{Cost}_{\text{scenario\_1}} \times p_1(\text{Case II}) + \text{Cost}_{\text{scenario\_4}} \times p_4(\text{Case II})$$
$$= \frac{h_{\text{unit}}(h_{\text{unit}}+1)}{2} + 2 + n_{\text{leader\_units}} - p_1(\text{Case II})$$

Figure 4.20 illustrates the average computation costs of leave for the CKC in both cases with the growth of group size, where the operation unit has 32 members.

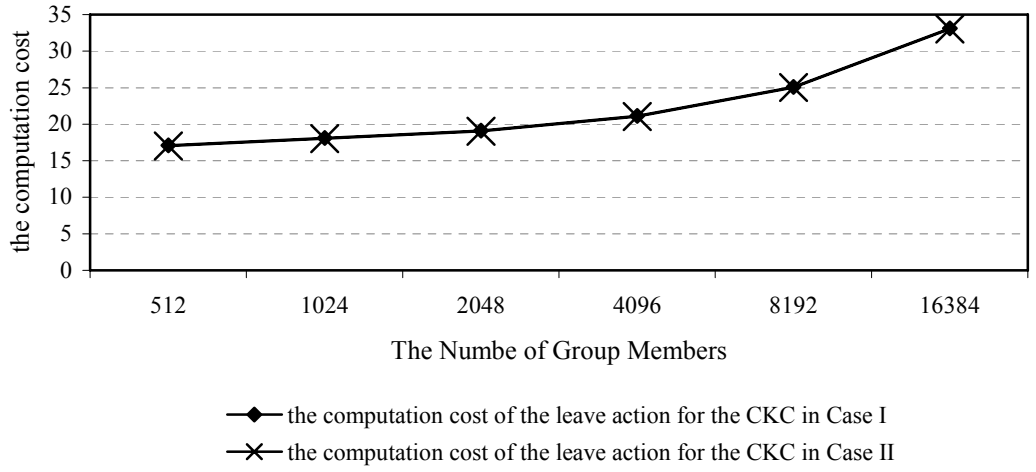


Figure 4.20 The computation cost of the leave action for the CKC in Cases I and II

From Figure 4.20, it can be observed that the costs for both cases are the same and increase with variations in group size. In both cases, the scenario of member-leave-from-a-member-unit makes a major and similar contribution to the total computation cost of the leave action, since the majority of members are in member units and the most frequently occurring leave action is the member-leave-from-a-member-unit.

A comparison of the computation cost of the leave action for HGKM (Cases I and II), LKH and OFT is shown in Figure 4.21.

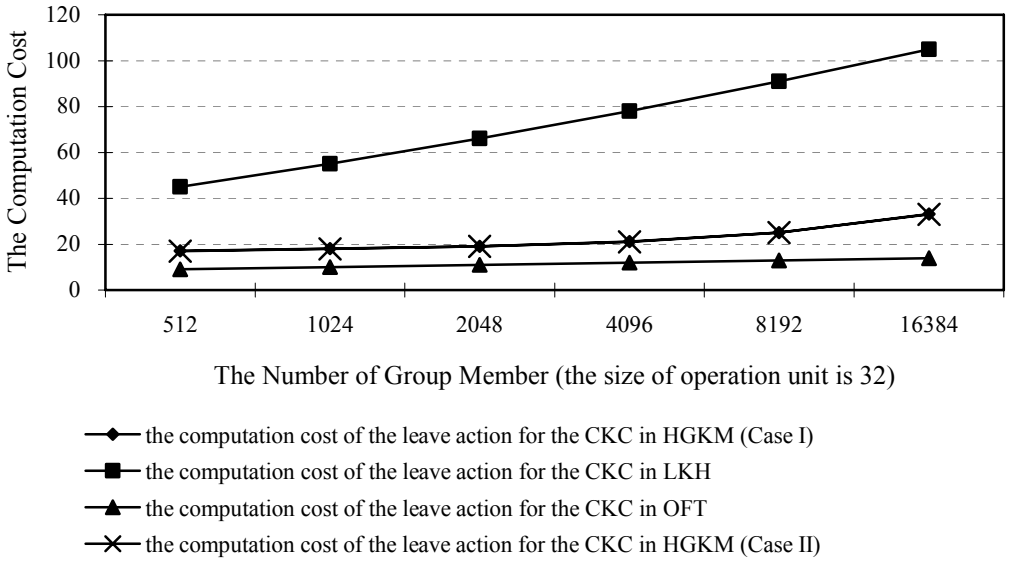


Figure 4.21 The computation cost of the leave action for the CKC

From Figure 4.21, it can be observed that OFT has the smallest computation cost of the leave action because it applies the one-way function to compute the intermediate supporting keys in the key tree. However, this computation efficiency is at the cost of security; OFT is susceptible to collusion attacks, where ex-group members can cooperate to calculate the current group key by applying their outdated keys. HGKM can also achieve computation efficiency for the CKC, although the cost is a little higher than that of OFT. Nevertheless, HGKM has no security loophole that can be subject to collusion attacks. The average computation cost of the leave action for the CKC in HGKM changes slowly with the growth of group size. LKH has the highest computation cost, two to three times of that of HGKM. Moreover, this cost increases exponentially with the growth of group size.

(ii) The computation cost of the leave action for members

Table 4.9 summarizes the computation cost of the leave action for members in HGKM during the rekeying process, where the formulas are from the section 2.2 and 4.3.5.

Table 4.9 The computation cost of the leave action for members

Group key management approaches		The computation cost		
		Members in directly-affected operation unit	Newly-chosen leader of the affected member unit	Members outside the directly-affected operation unit
HGKM	Scenario (i) A member leaves a member unit	$h_{\text{unit}}+1$	0	1
	Scenario (ii) A leadership candidate leaves	$h_{\text{unit}}+1$	0	1
	Scenario (iii) A leader leaves the group and a leadership candidate is available	$h_{\text{unit}}+1$	2	1
	Scenario (iv) A leader leaves the group and no leadership candidate is available	$h_{\text{unit}}+1$	0	1
LKH		h		
OFT		h		

h_{unit} : the height of the key tree for operation unit

h : the height of the group key tree

In Table 4.9, it can be observed that there are three types of computation cost of the leave action for different members. Since HGKM applies micro-key management within the operation unit and the members are treated differently, there are three kinds of members during rekeying, as follows:

- members in the directly-affected operation unit;
- the newly-chosen leader for the directly-affected member unit; and
- members outside the directly-affected operation unit.

Members within the directly-affected operation unit have the highest computation cost of the leave action. This cost is proportional to the height of the key tree for the operation unit. Due to the small size of the operation unit, this computation cost is still low compared to that of LKH and OFT. The newly-chosen leader of the directly-affected operation unit only receives two rekeying messages during the rekeying procedure. The members outside the directly-affected leader unit only receive one single rekeying message targeted to them. In contrast, in LKH and OFT, group members receive h rekeying messages where h equals the height of the group key tree.

Figure 4.22 illustrates the computation cost of the leave operation for the members in HGKM, LKH and OFT. We assume that the size of the operation unit is 32.

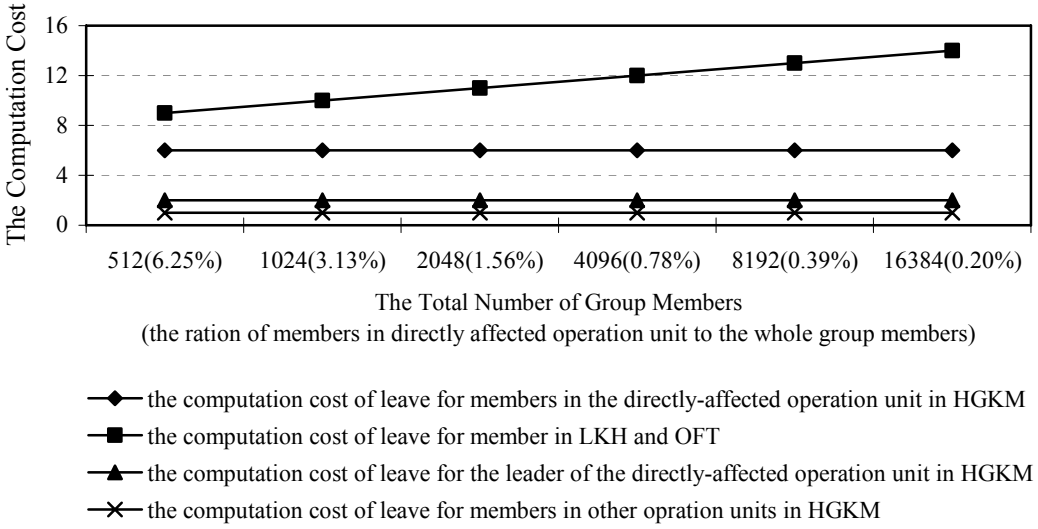


Figure 4.22 The computation cost of the leave operation for members

From Figure 4.22, we can observe that, for members, HGKM has an advantage over LKH and OFT in relation to the computation cost of the leave operation. In HGKM, due to the application of micro-key management in the operation units, major rekeying is performed within the directly-affected operation unit where a member leaves the group. The computation cost of the leave action for members in the directly-affected operation unit is still low, because the size of the operation unit is small. Moreover, members outside the directly-affected operation unit only receive one rekeying messages - an improvement compared to those in LKH and OFT.

In Figure 4.22, it can be seen that HGKM reduces the computation cost for members and benefits the resource-limited mobile devices. When the size of the operation unit is determined in HGKM, the computation cost of the leave action becomes a constant value and independent from the group sizes. This helps members manage the capacity of their mobile devices for participating in the group

applications. In contrast, the computation cost of the leave action for members in LKH and OFT is several times that of HGKM. Moreover, the computation cost for LKH and OFT increases with the growth of the group size.

In summary, based on the above analysis and comparison, HGKM can achieve better computation efficiency for members during rekeying for the leave operation than LKH and OFT, especially for the majority of members who are outside of the directly-affected operation unit.

4.4.2.3 Summary

In this section, we have analyzed and evaluated the computation cost for both the CKC and members in HGKM against LKH and OFT. We have determined that HGKM can attain greater computation efficiency in the rekeying process for both join and leave operations. The contributions of HGKM to the computation efficiency can be summarized as follows.

- Due to the application of micro-key management within the operation unit, the rekeying operation is confined within a small area. The computation cost for the CKC is therefore reduced.
- The small computation cost for members can reduce the workload on light-weight mobile devices, making HGKM suitable for the cellular wireless network.
- The constant value of computation cost for members in HGKM facilitates a user to judging its computation power to decide whether it can afford to join a group application.

Table 4.10 summarizes the computation cost of the join and leave operations for both the CKC and members in HGKM.

Table 4.10 The computation cost of the join and leave operations for HGKM

		Computation cost
join	The CKC	$\frac{(h_{\text{unit}}+1)(h_{\text{unit}}+2)}{2}+2-\frac{n_{\text{member\_in\_leader\_units}}}{n_{\text{total\_group\_members}}}$
	Members in the directly- affected operation unit	$h_{\text{unit}}+1$
	Members outside the directly-affected operation unit	1
leave	The CKC in Case I	$\frac{h_{\text{unit}}(h_{\text{unit}}+1)}{2}+5+n_{\text{leader\_units}}-4\times p_1(\text{Case I})-$ $5\times p_2(\text{Case I})$
	The CKC in Case II	$\frac{h_{\text{unit}}(h_{\text{unit}}+1)}{2}+2+n_{\text{leader\_units}}-p_1(\text{Case II})$
	Members in the directly-affected operation unit	$h_{\text{unit}}+1$
	Newly-chosen leader of the directly- affected member unit	2
	Members outside the directly-affected operation unit	1

h_{unit} : the height of operation unit;
$n_{\text{member\_in\_leader\_units}}$: the total number of members in the leader unit;
$n_{\text{total\_group\_members}}$: the total number of members in the gorup;

$p_1(\text{Case I})$: the probability of scenario (i) in Case I, $p_1(\text{Case I})=\frac{n_{\text{member\_in\_member\_units}}}{n_{\text{total\_group\_members}}}$;

$p_2(\text{Case I})$: the probability of scenario (ii) in Case I, $p_2(\text{Case I})=\frac{n_{\text{leadership\_candidates}}}{n_{\text{total\_group\_members}}}$;

$p_1(\text{Case II})$: the probability of scenario (i) in Case II, $p_1(\text{Case II})=\frac{n_{\text{member\_in\_member\_units}}}{n_{\text{total\_group\_members}}}$.

4.4.3 Key Storage Cost

Key storage cost measures the number of keys stored on both the CKC and members' mobile devices. In HGKM, as a result of applying micro-key management, a member is assigned to a small operation unit where a hierarchical key structure is built for key management. A member in the operation unit thus needs to keep a set of keys from its leaf node along the path to the root node. The number of stored keys is $h_{\text{unit}}+1$, where h_{unit} is the height of the key tree for the operation unit. Besides these keys, the member also needs to store the group traffic encryption key (GTEK) and the cell traffic encryption key (CTEK) for participating in the group application. Therefore, the total number of keys a member needs to store is $h_{\text{unit}}+3$. If a member is designated as a leader, it needs to store an extra key - the unit key of the corresponding member unit. Therefore, the number of keys stored by a leader is $h_{\text{unit}}+4$.

In terms of LKH and OFT, when these two approaches are applied in the wireless cell, a member also needs to store a set of keys from its leaf node along the path to the root node, plus the GTEK and CTEK. Thus, the total number of stored keys is $h+2$ (the key for the root node can be served as CTEK), where h is the height of the group key tree for LKH and OFT.

On the CKC side, in HGKM, all operation units have the same fixed size, therefore, the total number of keys stored is:

$$n_{\text{unit}} \times n_{\text{keys\_in\_unit}}$$

where n_{unit} is the number of operation units and $n_{\text{keys\_in\_unit}}$ is the number of keys

stored in a operation unit. If a binary tree is applied in the operation unit, the number of keys stored in the operation unit is:

$$n_{\text{keys\_in\_unit}} = 1+2+4+8+...+s_{\text{operation\_unit}} = 2s_{\text{operation\_unit}} - 1$$

where $s_{\text{operation\_unit}}$ is the size of the operation unit.

Therefore, the total number of keys stored on the CKC is:

$$n_{\text{unit}} \times n_{\text{keys\_in\_unit}} = \frac{s}{s_{\text{operation\_unit}}} \times (2s_{\text{operation\_unit}} - 1)$$

In LKH and OFT (assuming the binary tree is also applied), the number of keys stored by the CKC is:

$$1+2+4+8+...+n = 2s-1$$

where s is the size of group.

Table 4.11 tabulates the key storage cost for HGKM, LKH and OFT.

Table 4.11 The key storage cost for HGKM, LKH and OFT

Group key management	CKC	Group user	
		Leader	Member/leadership candidate
HGKM	$\frac{s}{s_{\text{operation\_unit}}} \times (2s_{\text{operation\_unit}} - 1)$	$h_{\text{unit}} + 4$	$h_{\text{unit}} + 3$
LKH	$2s-1$	$h+2$	
OFT	$2s-1$	$h+2$	

$s_{\text{operation\_unit}}$: the size of operation unit in HGKM

s : the size of group

h_{unit} : the height of key tree for the operation unit in HGKM

h : the height of key tree for LKH and OFT

In Table 4.11, it can be observed that, in HGKM, the key storage cost from the CKC's perspective depends on two factors, the size of the group and the size of operation unit. From the member's perspective, the key storage cost for HGKM is only proportional to the height of the key tree for the operation unit. If the size of the operation unit can be determined, the key storage cost for members is a constant value that facilitates the members to manage key storage space. In contrast, in terms of LKH and OFT, for the CKC, the key storage cost is proportional to the group size. The group size is larger than that of the operation unit. For members, the key storage cost is proportional to the height of the group key tree, which is decided by the group size.

Figures 4.23 and 4.24 illustrate the key storage costs from both the CKC's and members' perspective in HGKM, LKH and OFT. We assume the size of the operation unit is 32 in HGKM.

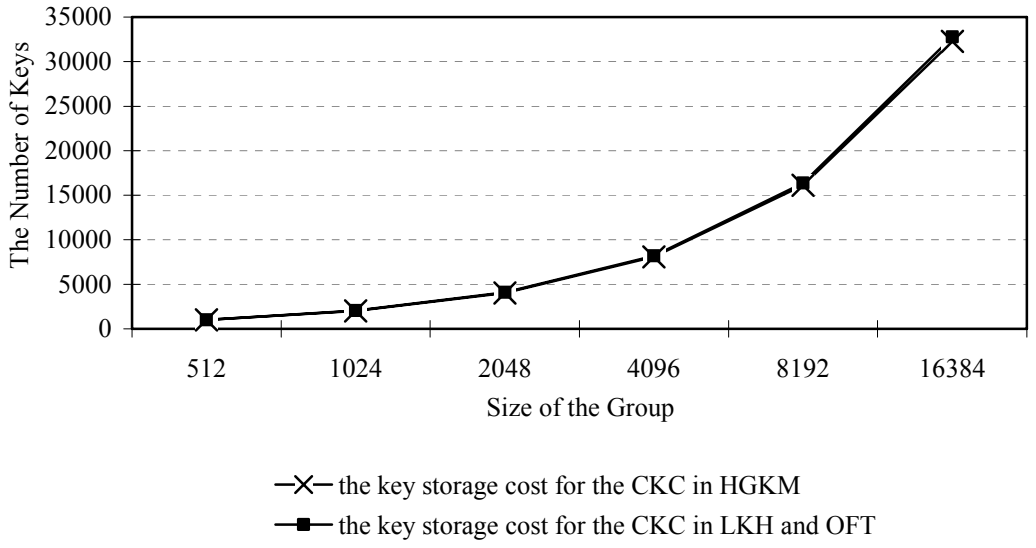


Figure 4.23 The key storage cost for the CKC in HGKM, LKH and OFT

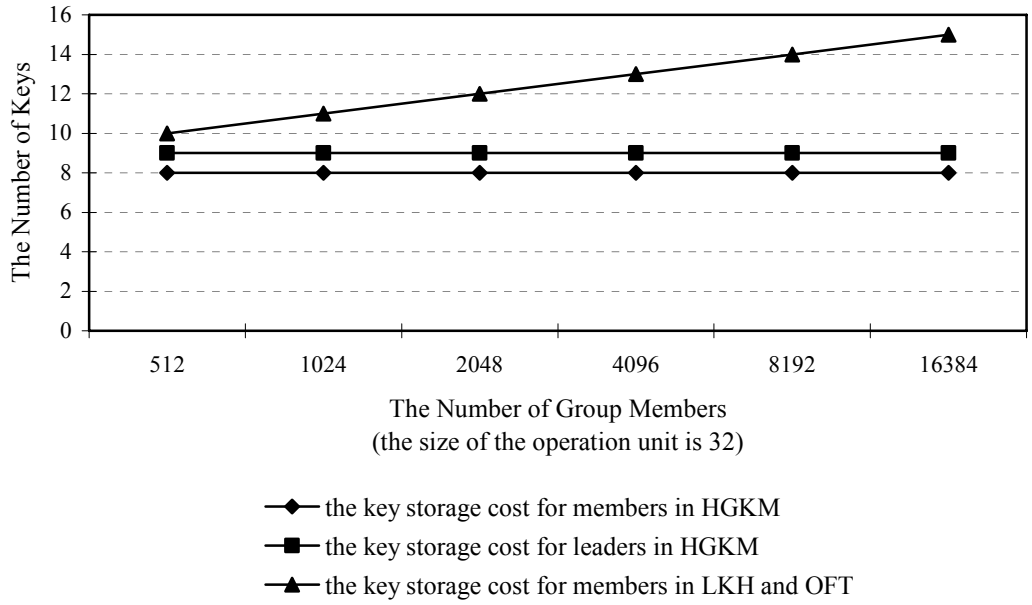


Figure 4.24 The key storage cost for members in HGKM, LKH and OFT

From Figures 4.23 and 4.24, it can be observed that, for the CKC, the key storage costs are quite similar in all three approaches as they all apply a hierarchical structure. For members, in relation to key storage cost, HGKM has the best performance. In Figure 4.24, when the size of the operation unit is determined, the key storage cost for HGKM becomes a constant value and independent from the group size. This benefits users to manage the key storage cost on mobile devices. In contrast, the key storage cost for members in LKH and OFT logarithmically increases with group size growth.

4.5 Summary

In order to tackle the problems of operational efficiency in wireless group key management, we have proposed a novel group key management approach: hybrid group key management (HGKM) in this chapter. This approach is specifically designed for the cellular wireless network. The major contribution of HGKM is that HGKM performs micro-key management within a small area known as operation unit. In HGKM, group members are organized into a number of operation units for the purpose of key management. Based on these operation units, micro-key management is performed. The features that allow HGKM to achieve operational efficiency can be summarized as follows:

- key management can be restricted within the small scope of the operation unit to reduce operational costs from the perspectives of communication, computation and key storage;
- HGKM combines the features of centralized and distributed key management approaches to allow members to involve in the key management in order to reduce the operational cost of the CKC during the rekeying process; and
- HGKM has a simple and built-in reliable message delivery scheme based on the operation unit to provide reliable transmission of keying materials.

Based on the analysis, evaluation and comparison presented in this chapter, we conclude that HGKM can improve operational performance from the perspectives of communication, computation and key storage. HGKM offers an efficient and

practical wireless group key management approach for the cellular wireless network. Table 4.12 summaries the operational costs for HGKM.

In the next chapter, we propose another group key management approach to address the operational efficiency problem: multiple-membership changes.

Table 4.12 The operation cost for HGKM

HGKM		Operation cost
Communication	Join (CKC)	3
	Leave (CKC)	$2+\frac{1}{s_{\text{operation\_unit}}}$
Computation	Join (CKC)	$\frac{(h_{\text{unit}}+1)(h_{\text{unit}}+2)}{2}+2-\frac{n_{\text{member\_in\_leader\_units}}}{n_{\text{total\_group\_members}}}$
	Join (member)	Members in the directly-affected operation unit: $h_{\text{unit}}+1$
		Members outside of the directly-affected operation unit: 1
	Leave (CKC)	Case I: $\frac{h_{\text{unit}}(h_{\text{unit}}+1)}{2}+5+n_{\text{leader\_units}}-4\times\frac{n_{\text{member\_in\_member\_units}}}{n_{\text{total\_group\_members}}}-5\times\frac{n_{\text{leadership\_candidates}}}{n_{\text{total\_group\_members}}}$
		Case II: $\frac{h_{\text{unit}}(h_{\text{unit}}+1)}{2}+2+n_{\text{leader\_units}}-\frac{n_{\text{member\_in\_member\_units}}}{n_{\text{total\_group\_members}}}$
	Leave (member)	Members in the directly-affected operation unit: $h_{\text{unit}}+1$
		Newly-chosen leader of the directly-affected member unit: 2
		Members outside the directly-affected operation unit: 1
Key storage	CKC	$\frac{s}{s_{\text{operation\_unit}}}\times(2s_{\text{operation\_unit}}-1)$

	member	Leaders: $h_{\text{unit}} + 4$
		Leader candidates and members in member units: $h_{\text{unit}} + 3$

$s_{\text{operation\_unit}}$: the size of operation unit in HGKM

h_{unit} : the height of the key tree for the operation unit in HGKM

$n_{\text{member\_in\_leader\_units}}$: the total number of members in the leader units

$n_{\text{total\_group\_members}}$: the total number of members in HGKM

$n_{\text{leader\_units}}$: the number of operation units in HGKM

$n_{\text{member\_in\_member\_units}}$: the number of members in the member units

$n_{\text{leadership\_candidates}}$: the number of leadership candidates in HGKM

s: the size of the whole group

Chapter 5

Membership-Oriented Key Management

In the majority of group applications, a service provider may offers several secure group applications concurrently. For example, an IPTV station may provide several channels to users, such as news, sports, movies and finance, while a user can subscribe to several channels at the same time. A user may thus hold concurrent multiple memberships. Multiple-membership is a common scenario in group applications. However, existing group key management approaches do not consider this situation. Under the existing approaches, a key management structure is associated with a group application. This means a separate key tree needs to be established for each group application. A user with multiple memberships has to register with several separate key trees in order to participate in the corresponding group communication. When multiple-membership changes, several key trees are affected. The operational costs of key management are expensive as rekeying is

performed in several separate key trees. The current approaches are thus inefficient, especially in the resource-limited wireless networks, where every effort needs to be made to reduce operational costs during key updating. In order to address this efficiency problem, a group key management approach, membership-oriented key management (MOKM) [Y. Chen & Wang et al., 2006; Wang & Le, 2005a; Wang & Le, 2005b; Wang & Le, 2007b], is proposed in this chapter. The major feature of MOKM is to reorganize the key management structure so that it is based on user membership. MOKM eliminates the redundancy of registrations in several key management structures and facilitates the key management operations during multiple-membership changes. In MOKM, regardless of the number of memberships a user owns, the user is assigned to one and only one key management structure. This allotment reduces the number of key management structures involved in the rekeying procedure.

The remainder of this chapter is organized as follows. Section 5.1 introduces the structure and operations of MOKM. The performance of MOKM is analyzed and evaluated in section 5.2, where the operational efficiency of MOKM in communication, computation and key storage is demonstrated by comparing it with an application-oriented key management approach. Finally, section 5.3 summarizes the chapter.

5.1 Membership-Oriented Key Management

In this section, we introduce MOKM. First, we investigate the logical structure of MOKM in subsection 5.1.1. Then, from section 5.1.2 to section 5.1.4, we discuss the key management operations in MOKM based on three membership change actions: join, leave and membership switch (switch).

5.1.1 Logical Structure of Membership-Oriented Key Management

In a traditional group key management scheme, key management operations are based on the group application: a key tree needs to be associated with a group application to form a one-to-one relationship in order to perform key management. However, this application-oriented key management mechanism prevents traditional group key management approaches from processing multiple-membership changes efficiently, as several separate key trees are affected when multiple-membership changes occur. In order to address the issue of multiple-membership changes, group key management needs to be changed from an application-oriented approach to a membership-oriented approach. The key management structure needs to be organized according to the user membership, rather than the group application. We have developed MOKM to address the problem of multiple-membership changes.

In MOKM, the key management structure, called *key-group*, is independent of the group applications. Each key-group is established based on the user membership. Members with the same membership(s) reside in the same key-group for the purpose of key management. MOKM has two types of key-groups: the single-membership

key-group and multiple-membership key-group. The single-membership key-group holds users with a single membership, while the multiple-membership key-group accommodates users having multiple memberships. Figure 5.1 illustrates an example of this new key management structure.

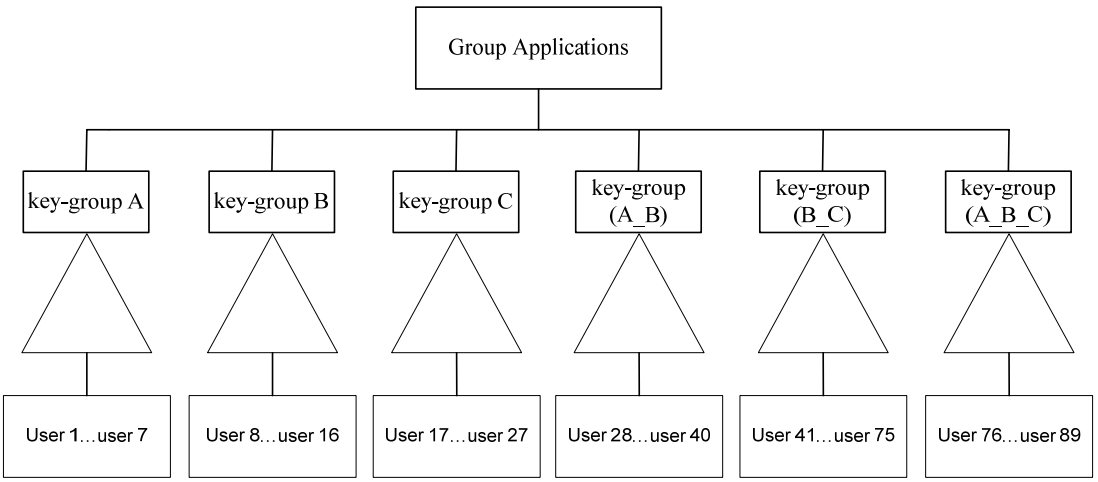


Figure 5.1 The key management structure in MOKM

In Figure 5.1, a service provider offers three group applications: A, B and C. Users can subscribe to group A, B, C or a combination of (A_B), (B_C) and (A_B_C). The number of combinations of the applications is determined by the number of group applications supplied by the service provider. The maximum number of combinations with *n* group applications is $\sum_{i=2}^{n} C_n^i$, where *n* is the number of group applications offered by the service provider. However, the number of combinations in the real case is decided by the service provider and may be less than the maximum. In this example, the service provider provides three combinations: A_B, B_C and A_B_C. Members who subscribe to these three combinations have

multiple memberships. Multiple-membership key-groups, A_B, B_C and A_B_C, as shown in Figure 5.1, need to be created for these users. When a user joins the service, it is assigned into one and only one key-group based on its membership(s). For example, a user who only subscribes to group application A is assigned to key-group A and a user subscribing to group application A and B is assigned into key-group (A_B).

Figure 5.1 also illustrates MOKM's three-layered structure. This consists of a group application level, a key-group level and a group member level.

- The group application level accommodates the Traffic Encryption Keys (TEKs). Each TEK is associated with a separate secure group application. In Figure 5.1, there are three TEKs in the group applications level, these being the TEKs for group applications A, B and C respectively.
- At the key-group level, key-groups are established according to individual or combinations of group applications. This level is thus responsible for the generation of the multiple-membership key-groups. The number of multiple-membership key-groups is determined by the number of combinations of group applications offered by the service provider.
- The lowest level consists of group members. In this level, regardless of the number of memberships, a member is assigned into one and only one appropriate key-group according to its membership(s). Within each key-group, any group key management approach (centralized or distributed) can be applied to organize group members for the purpose of key management.

In the following sections, we discuss key management operations in MOKM based on the three membership change actions: join, leave and switch. Without loss of generality, in this chapter, a hierarchical key tree is applied within the key-group because the hierarchical structure is one of the most powerful and efficient structure for key management.

5.1.2 Member Join

A user who wants to become a member of a group or a number of groups invokes the action of join. In the multiple-membership scenario, a user can subscribe to one or several group applications from the same service provider. TEK needs to be updated to ensure backward secrecy when a user joins a group. In MOKM, the join process can be accomplished following a three-step bottom-to-top procedure: user registration, key updating within the directly-affected key-group and TEK updating for all affected key-groups.

- Step 1: User registration

  In this stage, user u who wants to join the group application(s) sends a join request to the group key controller (GKC) which controls key management.

  $$\text{user } u \rightarrow \text{ GKC: \{group application(s) join request\}}$$

  After successful authentication, the GKC assigns user u to a key-group based on its membership(s). In order to enforce backward secrecy, the GKC needs to generate new supporting keys for the directly-affected key-group in which the incoming member resides. The GKC sends the new joining user these new keys

encrypted by a pair-wise key, k_{GKC_user}, known only to the GKC and the user.

$$GKC \rightarrow \text{user } u : \{\text{new keys which user } u \text{ is entitled to know}\}k_{GKC\_user(u)}$$

- Step 2: Key updating within the directly-affected key-group

  In step 2, the GKC updates the supporting keys within the directly-affected key-group to which the new joining member is assigned. The rekeying procedure is the same as that for LKH (described in section 2.2.1).

- Step 3: TEK updating for all affected key-groups

  After rekeying in the directly-affected key-group, the GKC updates the TEK(s) for all affected key-groups. For instance, in Figure 5.1, if user 1 joins application A, the GKC needs to generate a new TEK for application A (TEK_A') and send this new TEK to the key-group A, (A_B) and (A_B_C) because the members in these three key-groups also participate in group application A as well.

We provide an example to further explain the rekeying process of a multiple-membership join. For example, user 28 in Figure 5.1 wants to join group A and B. As user 28 subscribes to two group applications, it is assigned into key-group (A_B). In step 1, the GKC generates a set of new supporting keys to replace the affected ones in key-group (A_B) in order to ensure backward secrecy. The GKC sends these new supporting keys to user 28.

$$\text{user } 28 \rightarrow \text{GKC: } \{\text{request for join group application A and B}\}$$

$$GKC \rightarrow \text{user } 28\text{: } \{\text{newly-generated suporting keys in key-group (A\_B)}\}k_{GKC\_user(28)}$$

In step 2, the GKC updates the affected supporting keys in key-group (A_B).

$$GKC \Rightarrow \text{key-group (A\_B): } \{\text{newly-generated supporting keys}\}$$

Steps 1 and 2 are the typical rekeying process for the hierarchical key structure discussed in depth in section 2.2.1. Therefore, during rekeying, there are $h_{\text{key-group}}+1$ messages sent by the GKC and the number of keys encrypted by the GKC is

$$\frac{(h_{\text{key-group}}+1)(h_{\text{key-group}}+2)}{2}-1$$

where $h_{\text{key-group}}$ is the height of the key tree for the directly-affected key-group.

Finally, in step 3, the GKC updates the TEKs for all affected key-groups at the key-group level. In this example, key-groups A, B, A_B, B_C and A_B_C are affected by the addition of user 28. The GKC generates new TEKs for group applications A and B (i.e. TEK_A', TEK_B'). The GKC then generates a rekeying message in which new TEKs are encrypted by the root keys of these five key-groups (i.e. $k_\text{A}, k_\text{B}, k_{\text{A\_B}}, k_{\text{B\_C}}$ and $k_{\text{A\_B\_C}}$) respectively, and multicasts this rekeying message within the key-group level.

$$\begin{aligned}\text{GKC} \Rightarrow\ & \text{key-groups A, B, (A\_B), (B\_C) and (A\_B\_C)}: \\ & \{\{\text{TEK}_\text{A}'\}k_\text{A}, \{\text{TEK}_\text{B}'\}k_\text{B}, \{\text{TEK}_\text{A}', \text{TEK}_\text{B}'\}k_{\text{A\_B}}, \\ & \{\text{TEK}_\text{B}'\}k_{\text{B\_C}}, \{\text{TEK}_\text{A}', \text{TEK}_\text{B}'\}k_{\text{A\_B\_C}}\}\end{aligned}$$

During step 3, the GKC multicasts one rekeying message and the number of keys encrypted by the GKC is $\sum_{i=1}^{n_{\text{TEK}}(\text{affected})} i \times n_{\text{key-group}}(i)$, where $n_{\text{TEK}}(\text{affected})$ is the number of TEKs affected by the join operation, and $n_{\text{key-group}}(i)$ is the number of key-groups affected by *i* new TEK(s). In this example, three key-groups, key-group A, B and B_C, are affected by one new TEK (TEK_A') and two key-groups (i.e.

key-group (A_B) and (A_B_C)), are affected by two new TEKs (TEK_A' and TEK_B'). Therefore, the number of keys encrypted by the GKC in step 3 is:

$$\sum_{i=1}^{n_{\text{TEK}}^{(\text{affected})}} i \times n_{\text{key-group}}(i) = 1 \times 3 + 2 \times 2 = 5$$

In total, $h_{\text{key-group}} + 2$ rekeying messages are sent by the GKC during the rekeying process, where $h_{\text{key-group}}$ is the height of the key tree for the directly-affected key-group and the number of keys encrypted by the GKC is:

$$\frac{(h_{\text{key-group}} + 1)(h_{\text{key-group}} + 2)}{2} - 1 + \sum_{i=1}^{n_{\text{TEK}}^{(\text{affected})}} i \times n_{\text{key-group}}(i) .$$

5.1.3 Member Leave

In the multiple-membership scenario, the leave operation occurs when a member exits all group applications to which it subscribes. If a member does not leave all the group applications, we define this as a form of membership switch (switch), because the user is still a member of one or more group applications.

The leave operation can be invoked by a user who wants to leave an application or the GKC can evicts a user from group communications. Forward secrecy must be ensured to prevent the departing user from accessing future group communication. The rekeying procedure for the leave operation is similar to the join process. There are three steps: deregistering the member, key updating within the directly-affected key-group and TEK updating for all affected key-group(s).

- Step 1: Deregistering the member

  In this step, user *u* submits a leave request to the GKC to leave the group(s):

  $$\text{user } u \rightarrow \text{ GKC: \{leave request\}}$$

  After receiving this leave request, the GKC deletes the member from the corresponding group member list.

- Step 2: Key updating within the directly-affected key-group

  The GKC generates new supporting keys to replace the keys affected by the member departure. After key generation, the GKC multicasts these new supporting keys for key updating within the directly-affected key-group the member leaves. This rekeying process is the same as that in LKH if a hierarchical key structure is applied.

- Step 3: TEK updating for all affected key-group(s)

  Finally, in step 3, the GKC generates the new TEK(s) for the affected group application(s). After generation of the new keys, the GKC creates a rekeying message for all affected key-groups. This message contains the new TEK(s) encrypted by the root keys of the corresponding key-groups.

We provide an example to better illustrate the rekeying process for the leave action. If user 41 in Figure 5.1 wants to leave the group application B and C, in the first step, the GKC needs to deregister user 41 from the key-group (B_C).

In step 2, the GKC generates new supporting keys for the directly-affected key-group (B_C) and multicasts these new keys within it.

$$\text{GKC} \Rightarrow \text{key-group (B\_C)} : \{\text{new supporting keys for key updating}\}$$

Based on the rekeying process in LKH (discussed in section 2.2.1), in this step, $h_{\text{key-group}}$ rekeying messages are sent by the CKC, where $h_{\text{key-group}}$ is the height of the key tree for the directly-affected key-group. Meanwhile, the number of keys encrypted by the GKC is $\frac{h_{\text{key-group}}(h_{\text{key-group}}+1)}{2}$.

In the final step, the GKC generates the new TEKs, TEK_B' and TEK_C', for the directly-affected group applications B and C. The GKC generates the rekeying messages for the key-groups B, C, A_B, B_C and A_B_C affected by the TEK updating.

$$\begin{aligned}\text{GKC} \Rightarrow \text{key-groups B, C, (A\_B), (B\_C), (A\_B\_C)}: \\ \{\{\text{TEK}_\text{B}'\}k_\text{B}, \{\text{TEK}_\text{C}'\}k_\text{C}, \{\text{TEK}_\text{B}'\}k_\text{A\_B}, \\ \{\text{TEK}_\text{B}', \text{TEK}_\text{C}'\}k_\text{B\_C}, \{\text{TEK}_\text{B}', \text{TEK}_\text{C}'\}k_\text{A\_B\_C}\}\end{aligned}$$

During this step, the GKC multicasts a single rekeying message. The number of keys encrypted by the GKC is

$$\sum_{i=1}^{n_{\text{TEK}}(\text{affected})} i \times n_{\text{key-group}}(i),$$

where $n_{\text{TEK}}(\text{affected})$ is the number of TEKs affected by the leave operation, and $n_{\text{key-group}}(i)$ is the number of key-groups which are affected by *i* new TEK(s). In this example, three key-groups (key-group B, C and A_B) are affected by one new TEK (TEK_B') and two key-groups (key-group (B_C) and (A_B_C)) are affected by two new TEKs (TEK_B' and TEK_C'). Therefore, the number of keys encrypted by GKC in step 3 is:

$$\sum_{i=1}^{n_{\text{TEK}}(\text{affected})} i \times n_{\text{key-group}}(i) = 1 \times 3 + 2 \times 2 = 7$$

In summary, based on the above discussion, it can be observed that there are $h_{\text{key-group}}+1$ rekeying messages sent by the GKC during the rekeying process for the leave action, where $h_{\text{key-group}}$ is the height of the key tree for the directly-affected key-group. The total number of keys encrypted by the GKC is $\frac{h_{\text{key-group}}(h_{\text{key-group}}+1)}{2}+\sum_{i=1}^{n_{\text{TEK}}^{(\text{affected})}} i \times n_{\text{key-group}}(i)$.

5.1.4 Membership Switch

In a multiple-membership scenario, membership switch (switch) operation refers to a member changing its membership(s) among the group applications from the same service provider. Switch operation can be further classified into three types:

- Type I: a member adds one or several group applications into its current service package;
- Type II: a member quits one or several group applications from its current service package; and
- Type III: a member quits one or several group applications and joins one or several new group applications based on its current service package.

Although there are three different types of switch actions, the rekeying procedures for each of these are the same in MOKM, because, in all three scenarios, the switching member leaves the current key-group and is assigned into a new key-group according to its latest membership(s). Three steps of key updating are necessary for switch process: re-assigning the key-group, key updating within the two directly-affected key-groups and TEK updating for all affected key-groups.

- Step 1: Re-assigning the key-group

  The GKC reassigns a switching member into a new key-group based on the member's new membership(s). The GKC then sends the supporting keys of the new key-group to the member to ensure backward secrecy.

- Step 2: Key updating within the two directly-affected key-groups

  Two key-groups are directly affected by the switch: the current key-group and the new-assigning key-group. In this step, the GKC updates the supporting keys in both key-groups.

- Step 3: TEK updating for all affected key-groups

  The GKC generates new TEK(s) to replace the TEKs affected by the switch operation. This rekeying process is the same as that for the join and leave actions.

Because the same rekeying procedure is used for all three types of switch operations, we provide an example of type III switch action to further illustrate this rekeying process. For instance, in Figure 5.1, user 75 leaves group application C and joins group application A. Therefore, user 75 has membership of both application A and B after switch.

In step 1, the GKC re-assigns user 75 into key-group (A_B) based on its latest memberships and sends the newly-generated supporting keys in the key-group (A_B) to user 75.

$$\text{GKC} \rightarrow \text{user } 75 : \{\text{new generated supporting keys in key-group (A\_B)}\}k_{\text{GKC\_user(75)}}$$

Then, in step 2, the GKC updates the affected supporting keys for both directly-affected key-groups (A_B) and (B_C). The GKC performs rekeying of member join

in key-group (A_B) to which user 75 is reassigned. On the other hand, rekeying of member leave is operated in key-group (B_C) from where user 75 leaves.

GKC $\Rightarrow$ key-group (A_B): {newly-generated supporting keys for key updating}

GKC $\Rightarrow$ key-group (B_C): {newly-generated supporting keys for key updating}

Finally, in step 3, the GKC generates two new TEKs ($\mathrm{TEK_A}'$ and $\mathrm{TEK_C}'$) for the affected group applications A and C. After TEK generation, the GKC generates a rekeying message containing the new TEKs and multicasts it within the key-group level for TEK updating. In the example, key-groups A, C, A_B, B_C and A_B_C are affected by the switch.

$$\begin{aligned}\mathrm{GKC} \Rightarrow \text{key-group A, C, (A\_B), (B\_C), (A\_B\_C):}\\ \{\{\mathrm{TEK_A}'\}k_{\mathrm{A}}, \{\mathrm{TEK_C}'\}k_{\mathrm{C}}, \{\mathrm{TEK_A}\}k_{\mathrm{A\_B}},\\ \{\mathrm{TEK_C}'\}k_{\mathrm{B\_C}}, \{\mathrm{TEK_A}', \mathrm{TEK_C}'\}k_{\mathrm{A\_B\_C}}\}\end{aligned}$$

In summary, when a member switches its membership between group applications, it leaves its current key-group and joins a new key-group according to its latest membership(s). Rekeying for switch operation can therefore be regarded as a combination of one join and one leave operation. The number of rekeying messages sent by the GKC during the membership switch is:

$$h_{\text{key-group}}(\text{join}) + h_{\text{key-group}}(\text{leave}) + 2,$$

where $h_{\text{key-group}}(\text{join})$ and $h_{\text{key-group}}(\text{leave})$ is the height of the key tree for the key-group in the join and leave operation respectively.

The number of keys encrypted by the GKC during the membership switch is:

$$\frac{(h_{\text{key-group}}(\text{join})+1)(h_{\text{key-group}}(\text{join})+2)}{2}-1+\frac{h_{\text{key-group}}(\text{leaving})(h_{\text{key-group}}(\text{leaving})+1)}{2}+\sum_{i=1}^{n_{\text{TEK}}(\text{affected})} i\times n_{\text{key-group}}(i)$$
$$=(h_{\text{key-group}}(\text{join})+1)^2+\sum_{i=1}^{n_{\text{TEK}}(\text{affected})} i\times n_{\text{key-group}}(i)-1$$

where $n_{\text{TEK}}(\text{affected})$ is the number of TEKs affected by membership switch, and $n_{\text{key-group}}(i)$ is the number of key-groups affected by *i* new TEK(s).

5.2 Performance Analysis

As discussed at the beginning of this chapter, multiple-membership change is a serious performance issue in wireless group key management, because multiple key management structures are affected by multiple-membership change in the existing group key management approaches. In this section, we analyze and evaluate the operational efficiency of MOKM. The analysis and evaluation are conducted using three parameters: communication cost, computation cost and key storage cost, which were discussed in depth in section 3.2.2. We assume a hierarchical key tree is applied within the key-group to organize group members for the purpose of key management, because a hierarchical key structure is the most capable and efficient scheme for key management. We use the LKH approach as the benchmark for two reasons: first, it is foremost (in terms of popularity and efficiency) among the existing application-oriented group key management approaches, and second, it also uses a hierarchical key tree to achieve efficient key management.

5.2.1 Communication Cost

Communication cost (described in section 3.2.2) can be calculated using the number of rekeying messages sent by the GKC during the operations of join, leave and switch.

5.2.1.1 The Communication Cost of the Join Operation

In MOKM, regardless of the number of memberships of a new member, the joining member is assigned into one and only one key-group. Consequently, only one key-group is directly affected by the join operation. In addition, the main rekeying communication happens in the same key-group. Table 5.1 summarizes the communication cost of the join action for the GKC in MOKM and LKH. The formulas are taken from sections 5.1.2 and 2.2.1.

Table 5.1 The communication cost of the join operation for GKC

	Join operation
MOKM	$h_{\text{key-group}} + 2$
LKH	$\sum_{i=1}^{j} (h_{\text{group\_tree}}(i) + 1)$

$h_{\text{key-group}}$: the height of key tree for the directly affected key-group, which eqals $\log_2 n$, n is the szie of key-group
$h_{\text{group\_tree}}$: the height of group key tree for LKH
j: the number of group applications in which a user joins simultaneously ($j \geq 1$)

From Table 5.1, it can be observed that the communication cost of the join operation for the GKC in MOKM is proportional only to the height of the key tree for the directly-affected key-group that is determined by the number of members in

the directly-affected key-group and is independent of the number of group applications the user joins. In contrast, the communication cost of the join operation for LKH is proportional, not only to the number of members in the group application, but also to the number of group applications the new member joins.

We provide an example to further illustrate the communication cost of the join operation for the GKC in MOKM. We assume a service provider *s* offers three secure group applications: A, B and C. Users can subscribe to each of them or to a combination of (A and B), (B and C) or (A, B and C). We apply MOKM to reorganize the key management structure based on this scenario, as shown in Figure 5.2.

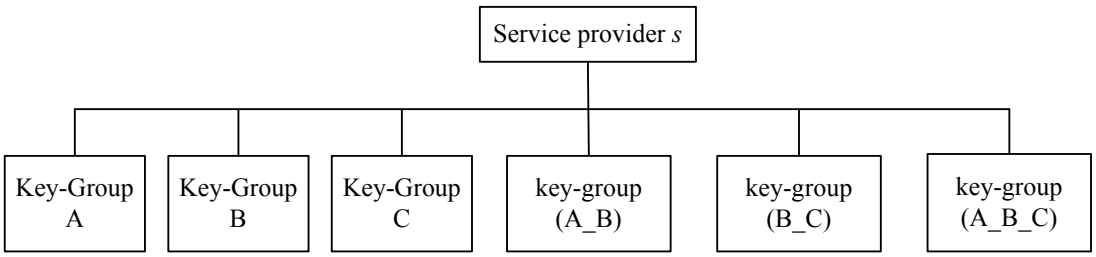


Figure 5.2 Structure of key groups

Without loss of generality, we apply a binary tree to establish the key tree in each key-group and assume that each key-group has the same number of group members, *n*. For LKH, the total number of members in each group application is:

- the number of members in group application A:

$$n(\mathrm{A}) = n_{\text{key-group}}(\mathrm{A}) + n_{\text{key-group}}(\mathrm{A\_B}) + n_{\text{key-group}}(\mathrm{A\_B\_C}) = 3n\ ;$$

- the number of members in group application B:

$$n(\mathrm{B}) = n_{\text{key-group}}(\mathrm{B}) + n_{\text{key-group}}(\mathrm{A\_B}) + n_{\text{key-group}}(\mathrm{B\_C}) + n_{\text{key-group}}(\mathrm{A\_B\_C}) = 4n \text{ ; and}$$

- the number of members in group application C:

$$n(C) = n_{\text{key-group}}(C) + n_{\text{key-group}}(\mathrm{B\_C}) + n_{\text{key-group}}(\mathrm{A\_B\_C}) = 3n .$$

In this example, we classify the join operation into three types:

Type I: a member joins a single application;

Type II: a member joins two applications; and

Type III: a member joins all three applications.

We assume that member *m* joins group application B to join a single application, joins applications A and C to join two applications and joins A, B and C to join all three applications. Figure 5.3 shows a comparison of communication costs of the join action for MOKM and LKH in this scenario.

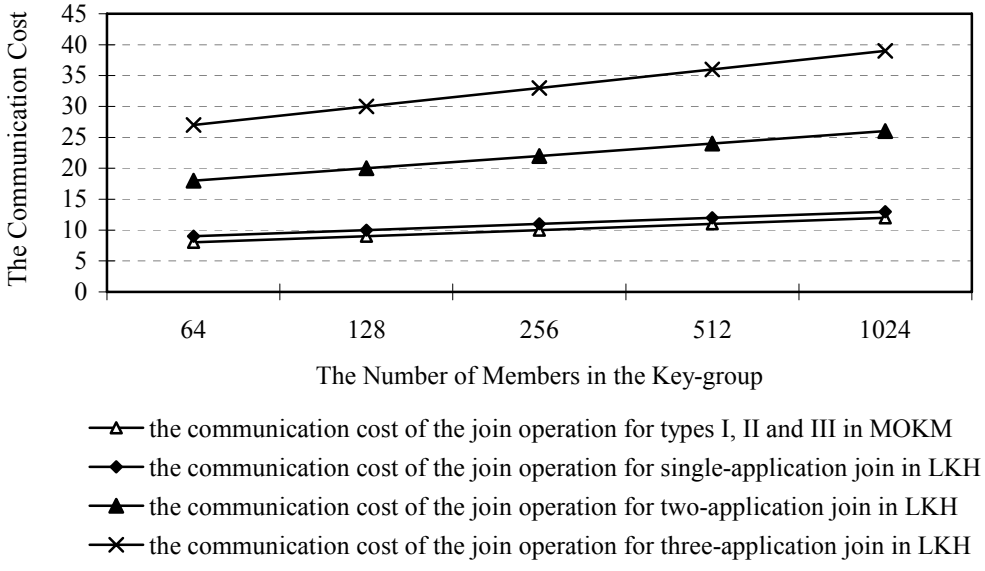


Figure 5.3 The communication costs of the join operation for MOKM and LKH

In Figure 5.3, it can be observed that, in MOKM, the communication cost of the join operation for all three types is the same. This is because the communication cost of the join operation in MOKM is independent of the number of group applications the user joins, and is only related to the height of the key tree for the directly-affected key-group. The height of the key tree is determined by the size of key-group. In this example, all key-groups have the same number of group members. Therefore, the communication costs for all three types of join action in MOKM are the same.

In Figure 5.3, like the single-application join, MOKM has a similar communication cost to that of LKH, because only one hierarchical key structure is affected by the join operation in both approaches. Regarding a multiple-application join, MOKM has an advantage over LKH. In Figure 5.3, for joining two applications, the communication cost for LKH is approximately two times of that of MOKM. In terms of joining three applications, the communication cost of LKH is approximately three times of that of MOKM. In MOKM, only one key-group affected by a multiple-application join. However, in LKH, several key trees are affected by a multiple-application join, because LKH is an application-oriented key management approach. When an increasing number of group applications are involved in the join operation, the performance of MOKM is better than that of LKH.

5.2.1.2 The Communication Cost of the Leave Operation

In MOKM, when a member leaves a group, MOKM performs key updating in three steps (discussed in depth in section 5.1.3). Table 5.2 tabulates the communication cost of the leave operation for MOKM and LKH.

Table 5.2 The communication cost of the leave operation for MOKM and LKH

	Leave operation
MOKM	$h_{\text{key-group}} + 1$
LKH	$\sum_{i=1}^{l} h_{\text{group-tree}}(i)$

$h_{\text{key-group}}$: the height of key tree for MOKM

$h_{\text{group-tree}}$: the height of key tree for LKH

l : the number of group applications from where the user leaves, $l \geq 1$

From Table 5.2, it can be observed that the result is similar to that of the join operation. The communication cost of the leave operation for MOKM is proportional only to the height of the key tree for the directly-affected key-group and is independent of the number of group applications involved in the leave operation. This is because MOKM reorganizes the key management structure according to the user membership(s) to ensure the member is only in one key-group. In contrast, the communication cost of the leave operation for LKH is proportional not only to the height of the group key tree, but also to the number of group applications involved in the leave operation.

We consider the same example shown in Figure 5.2 to analyze and evaluate the communication cost of the leave operation for MOKM and LKH. We demonstrate three different cases: (i) member *m* leaves application B (a departure from a single application), (ii) member *m* leaves applications A and C (a departure from two applications) and (iii) member *m* exits all three applications A, B and C (a departure from three group applications). Figure 5.4 illustrates the communication cost of the

leave operation with the growth in group size.

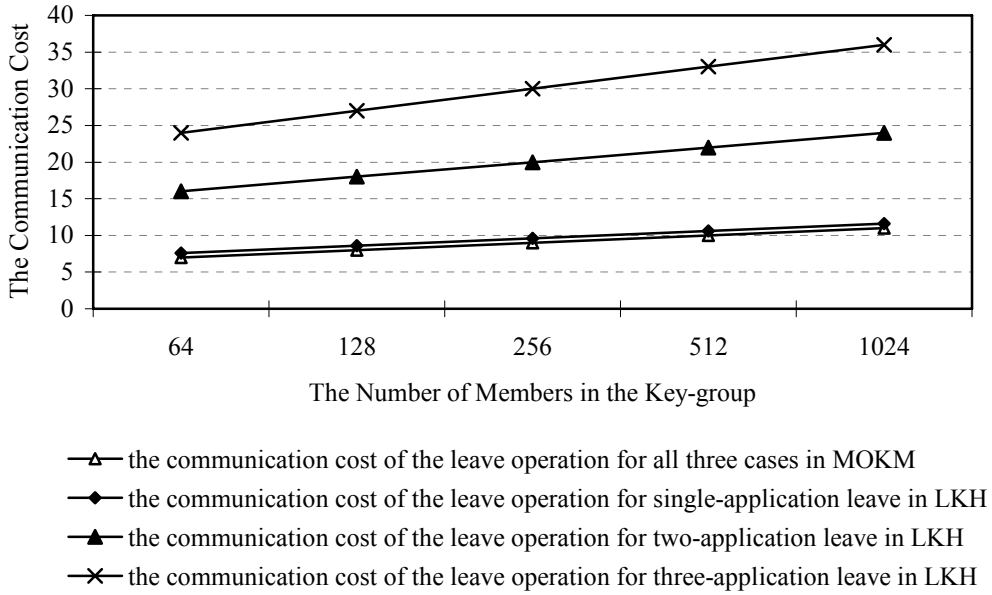


Figure 5.4 The communication cost of the leave operation for MOKM and LKH

In Figure 5.4, it can be observed that all three leave scenarios have the same communication cost in MOKM. This is because the communication cost of the leave operation for MOKM is proportional only to the height of the key tree for the directly-affected key-group (determined by the size of key-group), and is independent of the number of group applications involved in the leave operation. In MOKM, a group member is assigned to one and only one key-group according to its membership(s). The communication cost of the leave operation is therefore related only to that directly-affected key-group the member leaves. As a result, in this example, all three leave cases have the same communication cost of the leave operation for MOKM.

In terms of membership departure from a single-group application, MOKM and LKH have a similar communication cost (shown in Figure 5.4) because only one key tree is affected by the leave operation in both approaches. In terms of membership departure from multiple applications, there is still only one key-group affected in MOKM. In contrast, several separate key trees are affected in LKH. The total communication cost of the leave operation for a multiple-application scenario in LKH is the summation of the communication cost of each separate leave operation. In Figure 5.4, regarding the two-application departure, two separated key trees are affected by the leave operation. The communication cost for LKH is approximately two times that of MOKM. For the case involving departure from three group applications, the communication cost for LKH is approximately three times that of MOKM. Moreover, from Figure 5.4, it can be noticed that the gap in the communication cost of the leave operation between MOKM and LKH becomes increasingly wider with the increase in group size and the number of applications involved in the leave operation.

In conclusion, it appears that MOKM can achieve better communication efficiency than LKH during the rekeying process for the leave operation, particularly when multiple group applications are involved.

5.2.1.3 The Communication Cost of the Switch Operation

In MOKM, the three different types of the switch operation have the same rekeying procedure (described in section 5.1.4). Due to the re-organization of key management structures based on membership, only two key-groups are directly

affected by the switch operation within MOKM: the key-group the member leaves and the key-group to which the member is reassigned. The communication cost of the switch operation can therefore be treated as a combination of one join and one leave operation. In terms of the LKH approach, a membership switch involves join and leave operations in several group applications. Thus, several separate key trees are affected by the switch operation. The communication cost of the switch operation for LKH is the summation of the communication costs of these join and leave operations.

Table 5.3 The communication cost of the switch operation in MOKM and LKH

	Membership Switch
MOKM	$h_{\text{key-group}}(\text{join}) + h_{\text{key-group}}(\text{leaving}) + 2$
LKH	$\sum_{n=1}^{j}(h_{\text{group\_tree}}(n)+1) + \sum_{m=1}^{l} h_{\text{group\_tree}}(m) \qquad (j,l \geq 1)$

$h_{\text{key-group}}(\text{join})$: the height of the key tree for the key-group in MOKM, where member joins
$h_{\text{key-group}}(\text{leaving})$: the height of the key tree for the key-group in MOKM, from where member leaves
$h_{\text{group\_tree}}$: the height of group key tree for LKH
j : the number of group applications where new member joins
l : the number of group applications from where member leaves

From Table 5.3, it can be observed that the communication cost of the switch can be processed as a combination of the join and leave operations for both MOKM and LKH. In MOKM, only two key-groups are affected by the switch, regardless of the number of group applications involved. Therefore, the communication cost is proportional only to the size of these two directly-affected key-groups and independent of the number of group applications involved. In contrast, in LKH, several separate key trees are affected by the switch operation and the

communication cost for LKH is linearly proportional, not only to the size of the group, but also to the number of group applications involved.

We apply the same example shown in Figure 5.2 to calculate the communication cost of the switch operation for both MOKM and LKH. We assume an average of 1024 members in each key-group in Figure 5.2. The number of members in each group application for LKH is therefore:

- Group application A: $n(\mathrm{A}) = 3n = 3 \times 1024 = 3072$;
- Group application B: $n(\mathrm{B}) = 4n = 4 \times 1024 = 4096$; and
- Group application C: $n(\mathrm{C}) = 3n = 3 \times 1024 = 3072$.

We present that the following three scenarios as examples that involve one, two and three group applications for switch respectively.

- One group application involved: user *u* in group application B wants to subscribe also to application C.
- Two group applications involved: user *u* in key-group (A_B_C) wants to leave group applications A and C.
- Three group applications involved: user *u* in key-group (B_C) wants to leave group applications B and C, and join group application A.

Figure 5.5 illustrates the communication cost of the switch operation in this example.

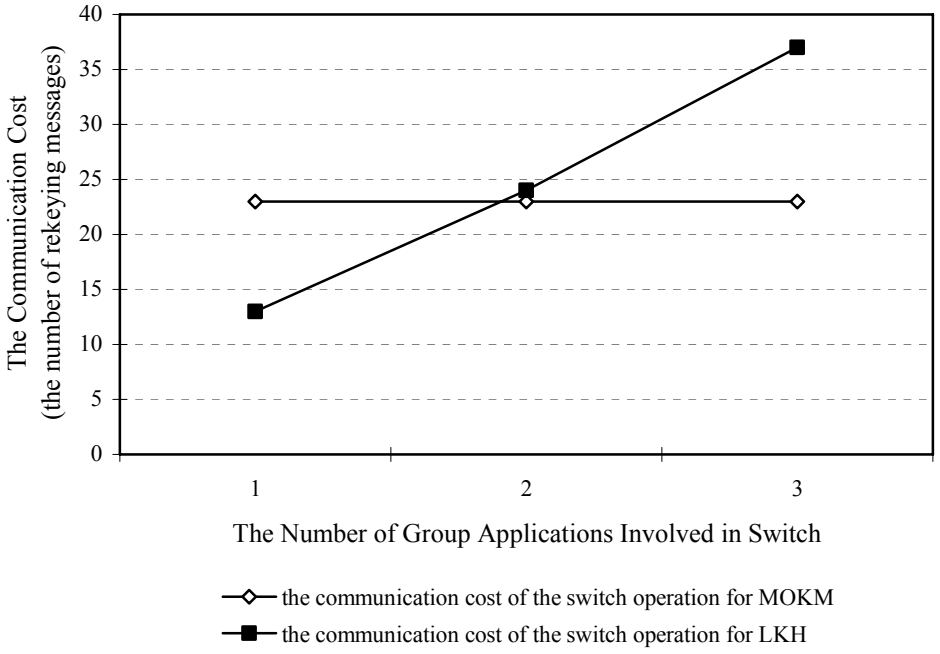


Figure 5.5 The communication cost of the switch operation for MOKM and LKH

In Figure 5.5, when one application is involved in the switch, LKH has a communication cost approximately 43% lower than MOKM, because only one key tree is affected in LKH and two key-groups are involved in MOKM. When multiple applications are involved in the switch, the communication cost of the switch operation for MOKM is the same as that of single application involved, because still only two key-groups are affected by multiple-application switch in MOKM and the number of members in each key-group remains the same in this example. In contrast, the communication cost of the switch operation for LKH increases sharply. As shown in Figure 5.5, when two group applications are involved in the switch, the communication cost for LKH is similar to that of MOKM. If three group applications are involved, the cost for LKH is about 60% more than that of MOKM. Moreover,

we can deduce that the communication cost of the switch operation for LKH rises linearly in relation to the increase in the number of group applications involved in the switch. In addition, the gap in the communication cost of the switch operation between MOKM and LKH becomes increasingly wider.

5.2.1.4 Summary

Based on the above analysis and evaluation of the communication costs for MOKM and LKH, MOKM has an advantage over LKH in communication costs when multiple group applications are involved. Table 5.4 summarizes the communication cost of the join, leave and switch operations for MOKM and LKH.

Table 5.4 The communication cost for MOKM and LKH

Operation	MOKM	LKH
Joining	$h_{\text{key-group}} + 2$	$\sum_{i=1}^{j} (h_{\text{group\_tree}}(i) + 1)$
Leaving	$h_{\text{key-group}} + 1$	$\sum_{i=1}^{l} h_{\text{group-tree}}(i)$
Membership switch	$h_{\text{key-group}}(\text{join}) +$ $h_{\text{key-group}}(\text{leaving}) + 2$	$\sum_{n=1}^{j} (h_{\text{group\_tree}}(n) + 1) + \sum_{m=1}^{l} h_{\text{group\_tree}}(m)$ $(j, l \geq 1)$

$h_{\text{key-group}}$: the height of the key tree for the directly affected key-group

$h_{\text{key-group}}(\text{join})$: the height of the key tree for the key-group in MOKM, where member joins

$h_{\text{key-group}}(\text{leaving})$: the height of the key tree for the key-group in MOKM, from where member leaves

$h_{\text{group\_tree}}$: the height of group key tree for LKH

j : the number of group applications where new member joins

l : the number of group applications from where member leaves

In MOKM, a membership-oriented key management structure is applied; that is, a key management structure is established based on user membership(s). A member is assigned to one and only one key-group based on its membership(s). This eliminates registration redundancy in several key management structures when a user wishes to participate in several group applications simultaneously. When MOKM is applied, either one or two key-groups are affected by the multiple-membership changes resulting from the operations of join, leave and switch. The communication cost for MOKM is proportional only to the number of members in the directly-affected key-group and is independent of the number of group applications involved. In contrast, the communication cost for LKH, an application-oriented group key management approach, is linearly proportional not only to the number of members in the affected group applications, but also to the number of group applications involved. In LKH, an increase of either group members or group applications involved in the switch operation causes an increase in communication cost. MOKM, in comparison, achieves greater communication efficiency for the scenario of multiple-membership change.

5.2.2 Computation Cost

Table 5.5 summaries the computation cost of the join, leave and switch operations for MOKM and LKH. A hierarchical structure is applied in both approaches. The formulas are taken from sections 5.1.2 to 5.14 and section 2.2.1.

Table 5.5 The computation costs for MOKM and LKH

operation	MOKM	LKH
Join	$\frac{(h_{\text{key-group}}+1)(h_{\text{key-group}}+2)}{2}-1+\sum_{i=1}^{n_{\text{TEK}}(\text{affected})} i\times n_{\text{key-group}}(i)$	$\sum_{i=1}^{j}\left(\frac{(h_{\text{group}}+1)(h_{\text{group}}+2)}{2}-1\right)$ $(j\geq 1)$
Leave	$\frac{h_{\text{key-group}}(h_{\text{key-group}}+1)}{2}+\sum_{i=1}^{n_{\text{TEK}}(\text{affected})} i\times n_{\text{key-group}}(i)$	$\sum_{i=1}^{l}\left(\frac{h_{\text{group}}(h_{\text{group}}+1)}{2}\right)$ $(l\geq 1)$
Switch	$(h_{\text{key-group}}(\text{join})+1)^2+\sum_{i=1}^{n_{\text{TEK}}(\text{affected})} i\times n_{\text{key-group}}(i)-1$	$\sum_{i=1}^{j}\left(\frac{(h_{\text{group}}+1)(h_{\text{group}}+2)}{2}-1\right)+\sum_{i=1}^{l}\left(\frac{h_{\text{group}}(h_{\text{group}}+1)}{2}\right)$ $(j,l\geq 1)$

$h_{\text{key-group}}$: the height of key-group in MOKM

$h_{\text{group\_tree}}$: the height of key tree for LKH

$n_{\text{TEK}}(\text{affected})$: the number of affected TEKs which need to be updated

$n_{\text{key-group}}(\text{i})$: the number of affected key-groups in MOKM, which are affected by i new TEK(s)

j : the number of group applications which are affected by joining operation in LKH

l : the number of group applications which are affected by leaving operation in LKH

From Table 5.5, we can observe that the computation cost for MOKM is proportional to the $O((h_{\text{key-group}})^2))$, while the computation cost for LKH is relative to the $O((h_{\text{group}})^2)$, where $h_{\text{key-group}}$ and h_{group} are the height of the key tree for the key-group in MOKM and the height of the group key tree in LKH respectively. The size of the key-group is smaller than that of the group application because the members in the key-group are a subset of all the members in the group application.

Thus, we can infer that the computation cost for MOKM is lower than that for LKH.

The example shown in Figure 5.2 can also be used to calculate the computation cost of the join, leave and switch operations. Without loss of generality, we make the following assumptions:

- Each key-group has an average of 1024 members; therefore, the number of group members in group application A, B and C is 3072, 4096 and 3072 respectively.
- In terms of join and leave operations,
  - user *u* joining and leaving application B represents a one-application case;
  - user *u* joining and leaving applications A and B represents two-application case; and
  - user *u* joining and leaving applications A, B and C represents three-application case.
- For switch, we consider one application involved when user *u*, in group application B wants to subscribe, in addition, to application C. User *u* in key-group (A_B_C) leaving applications A and C is treated as an example of a two-application changes. User *u* in key-group (B_C) leaving group applications B and C, and then joining group application A, is regarded as a three-application case.

Figures 5.6, 5.7 and 5.8 illustrate the computation cost pertaining to the membership changes of join, leave and switch with one, two and three group applications involved.

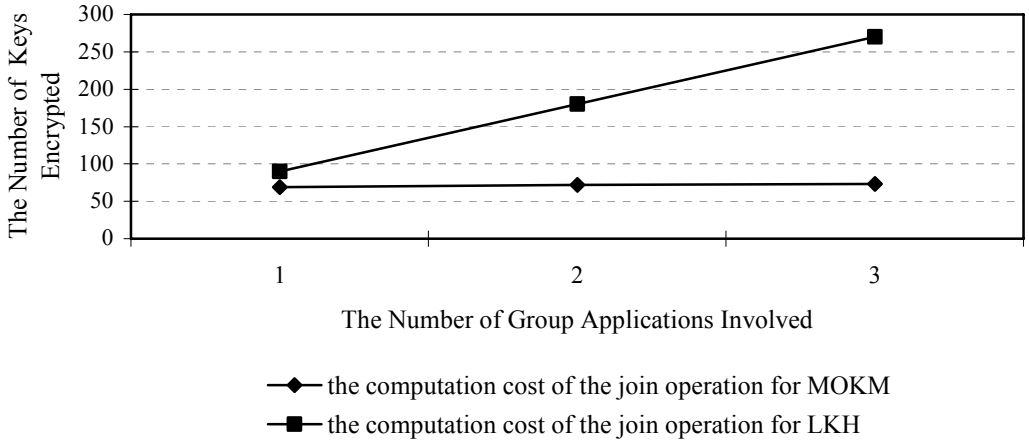


Figure 5.6 The computation cost of the join operation for MOKM and LKH

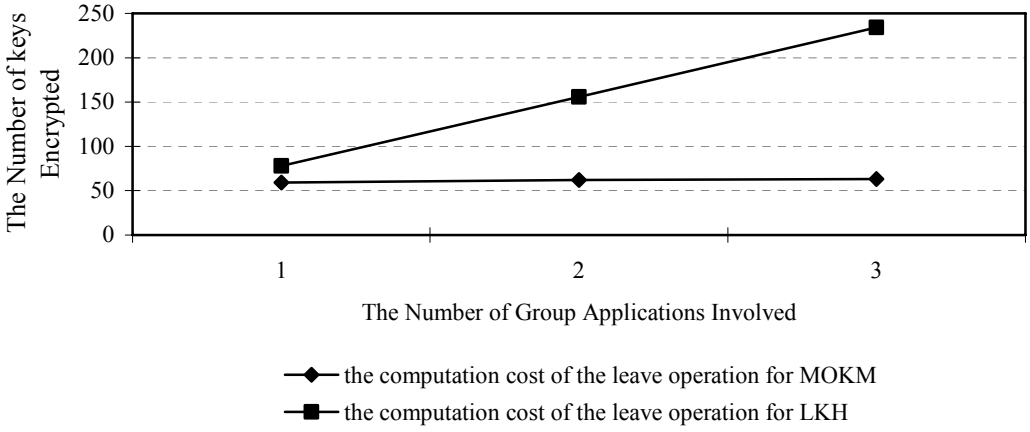


Figure 5.7 The computation cost of the leave operation for MOKM and LKH

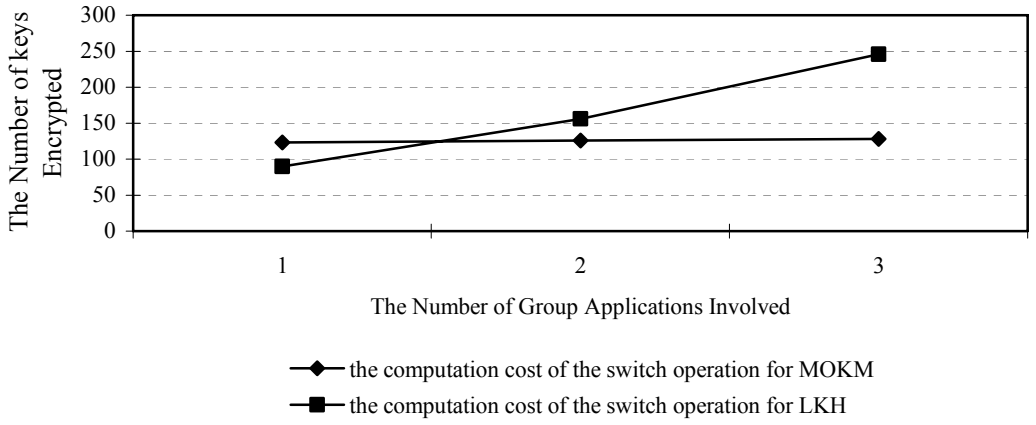


Figure 5.8 The computation cost of the switch operation for MOKM and LKH

From Figures 5.6, 5.7 and 5.8, it can be observed that MOKM has an advantage over LKH in computation cost in terms of all three membership changes (join, leave and membership switch), especially when multiple applications are involved.

For the join and leave operations, when one application is involved, only one key tree is affected in both approaches. MOKM and LKH therefore have a similar computation cost. When multiple applications are involved such as in the two- and three-application cases (shown in Figures 5.6, 5.7 and 5.8), MOKM has greater computation efficiency than LKH. By applying membership-oriented key management, in MOKM, only one key-group is affected by the join and leave operations. The computation cost is determined only by the number of members in the directly-affected key-group. In contrast, in LKH, two and three group key trees are affected by multiple-membership changes. The computation cost for LKH therefore increases linearly. Moreover, from Figures 5.6 and 5.7, it is reasonable to deduce that the gap in computation cost between MOKM and LKH also increases linearly in relation to the increase in the number of applications involved in the

membership changes.

In terms of the switch operation, when only one application is involved, the computation cost of LKH is less than that of MOKM, because LKH has one hierarchical structure affected while MOKM has two. When more group applications are involved in the switch, the computation cost for LKH increases linearly, because more key trees are affected and the computation cost for LKH is linearly proportional to the number of group applications involved. In contrast, only two key-groups are affected by a membership switch in MOKM, regardless of the number of group applications involved. Furthermore, similar to the join and leave operations, the gap in computation cost of the switch operation between MOKM and LKH increases as the number of group applications involved increases.

In conclusion, based on the above analysis, evaluation and comparison, MOKM can achieve greater computation efficiency than LKH for all three types of membership change (join, leave and switch operations). The benefit increases when multiple applications are involved.

5.2.3 Key Storage Cost

The key storage cost is measured by the number of keys stored on the GKC and the group members' mobile devices. In LKH, a traditional hierarchical structure key management approach, a member needs to remember a set of keys from its leaf node along the path to the root node, therefore, the number of keys stored for members is $h_{\text{group}}+1$, where h_{group} is the height of the group key tree. If a member joins multiple group applications, it needs to store several sets of keys; each set is for a

separate group application. Therefore, the total number of keys stored by the member is:

$$\sum_{i=1}^{j}(h_{\text{group}}+1)$$

where j is the number of group applications which the user joins.

For the GKC, in LKH, the total number of keys stored in the key tree is:

$$1+2+\cdots+n=2n-1$$

where n is the number of group members

In MOKM, if a hierarchical key tree is applied to the key-group, a member needs to store $h_{\text{key-group}}+1$ supporting keys, where $h_{\text{key-group}}$ is the height of the key tree for the key-group. In addition to $h_{\text{key-group}}+1$ supporting keys, a member also needs to store j TEKs, where j is the number of applications to which the member has subscribed. Therefore, the total number of keys a member needs to store in MOKN is $h_{\text{key-group}}+j+1$.

For the GKC, we assume that there are $n_{\text{applications}}$ TEKs, where $n_{\text{applications}}$ is the number of group applications. Thus, the number of keys stored on the GKC in MOKM is

$$\sum_{i=1}^{n(\text{key-group})}(2n_{\text{key-group}}(i)-1)+n_{\text{applications}},$$

where $n(\text{key-group})$ is the number of key-groups in MOKM and $n_{\text{key-group}}(i)$ is the number of members in ith key-group. The key storage cost for MOKM and LKH is summarized in Table 5.6.

Table 5.6 The key storage cost for MOKM and LKH

	GKC	member
MOKM	$\sum_{i=1}^{n(\text{key-group})} (2n_{\text{key-group}}(i)-1)+n_{\text{applications}}$	$h_{\text{key-group}}+j+1$
LKH	$\sum_{i=1}^{n_{\text{applications}}} (2n-1)$	$\sum_{i=1}^{j}(h_{\text{group}}+1)$

n : the number of members in the group application

$n_{\text{key-group}}(i)$: the number of members in ith key-group

$n_{\text{applications}}$: the number of group applications

$n(\text{key-group})$: the number of key groups in MOKM

j : the number of group applications to which a member subscribes ($j \geq 1$)

h_{group} : the height of the group key tree in LKH

From Table 5.6, it can be observed that, for the GKC, the key storage cost for both approaches is linearly proportional to the group size and the number of group applications involved. Regarding the number of keys stored by the members, in MOKM, a member is assigned to one and only one key-group. Therefore, the key storage cost is proportional only to the height of the key tree for the key-group in which the user resides and the number of group applications the member joins. In contrast, in LKH, the key storage cost for a member is the summation of the key storage cost for each group application that the member joins.

We consider the example shown in Figure 5.2 to calculate the key storage cost for both the GKC and the group members in MOKM and LKH. We assume the average number of members in each key-group is 1024. Figures 5.9 and 5.10 show the key storage cost for both the GKC and members in MOKM and LKH with an increase in the number of group applications.

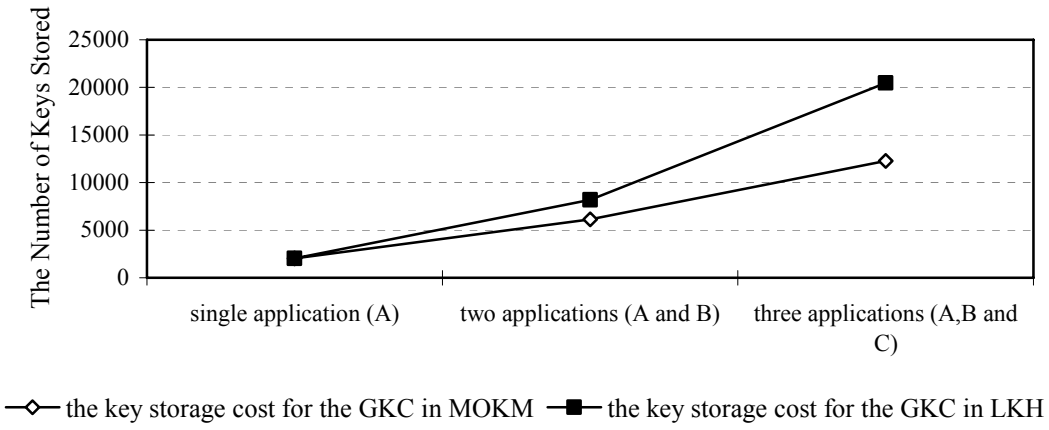


Figure 5.9 The key storage cost for the GKC in MOKM and LKH

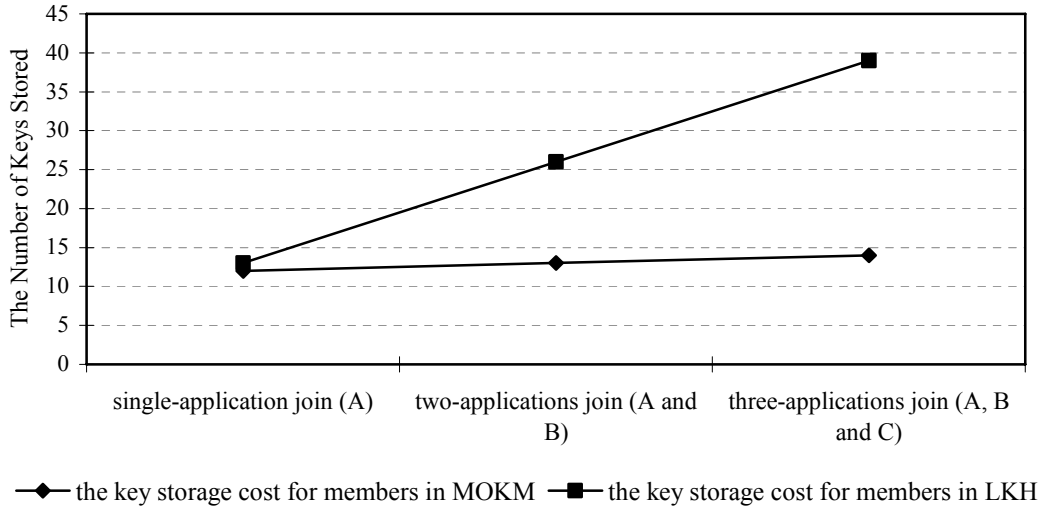


Figure 5.10 The key storage cost for members in MOKM and LKH

From Figure 5.9, for the GKC, when one application is involved, the key storage cost is almost the same for both MOKM and LKH because only one key tree is involved in both approaches. When multiple applications are considered, MOKM has a lower key storage cost than LKH because has no redundant registration. In MOKM, a member is assigned to one and only one key-group based on its

membership(s). Members do not need to register themselves with several key management structure in order to gain multiple memberships. Moreover, the size of the key-group is smaller than that of LKH because members in the key-group are a subset of all members in the group application. Both factors benefit the key storage cost of MOKM.

The key storage cost for members is similar to that of the GKC (see Figure 5.10). For members joining a single application, MOKM and LKH have similar key storage costs, because there is only one key management structure involved in both approaches. When multiple applications are involved, the key storage cost of LKH is higher than that of MOKM. In MOKM, a member is assigned to one and only one key-group, regardless of the number of group applications with which it registers. Therefore, the key storage cost for members is related only to the height of the key tree for that assigned key-group. In contrast, a member in LKH needs to store several sets of keys if it joins several group applications; each set of keys is for a separate group application. Therefore, the key storage cost for a member in LKH is proportional to both the height of group key tree (determined by the number of members in the group application) and the number of group applications the member joins. For members, the key storage cost in LKH increases as the number of group members and group applications increases.

5.3 Summary

In secure group applications, multiple-membership is a common phenomenon. Multiple-membership refers to a member simultaneously subscribing to several group applications supplied by the same service provider. Traditional group key management schemes such as LKH associate the key management structures with group applications. This association requires a separate key management structure to be established for each group application. An application-oriented key management approach leads to the redundant member registration in key management structures and inefficiency in the key management of multiple-membership changes, as several key structures are affected by group operations such as join, leave and switch.

In order to tackle this issue, we have proposed MOKM, a membership-oriented key management approach. The major contribution of MOKM is to reorganize the key management structure so that it is based on the user membership, where the key management structure, key-group, is independent of group applications. In MOKM, a member is assigned to one and only one key-group according to its membership(s). This new group key management approach eliminates the redundant member registration and facilitates key management for multiple-membership changes. During the multiple-membership changes, in MOKM, either one or two key-groups are affected. In contrast, several separate key management structures are affected in LKH. Based on the analysis, evaluation and comparison presented in this chapter, we conclude that MOKM can achieve greater efficiency in communication, computation and key storage than traditional application-oriented key management approaches

such as LKH.

MOKM is open structured and thus allows the application of various group key management schemes in key-group. In this chapter, for the purpose of comparison with LKH, we have applied a hierarchical structure to MOKM. However, the results obtained in this chapter can be applied to any application-oriented key management approach as key management structures in other application-oriented approaches also encounter the same inefficiency problems when multiple-membership changes are involved.

In this and the previous chapter, we have proposed two group key management approaches. Both are designed to tackle specific operational efficiency issues. In the next chapter, we integrate these two approaches with the proposed wireless group key management architecture to form a comprehensive group key management solution for the cellular wireless network. The next chapter investigates this solution in detail to demonstrate its operational efficiency and security.

Chapter 6

A Group Key Management Solution for the Cellular Wireless Network

In the previous chapters, we have proposed three approaches for the wireless group key management system: a group key management architecture for the cellular wireless network and two group key management approaches, hybrid group key management (HGKM) and membership-oriented key management (MOKM). The proposed wireless group key management architecture addresses the network-compatible issue of wireless group key management. By integrating the key management architecture with the underlying wireless network infrastructure, the group key management system can utilize the capacity of the underlying wireless network to provide scalability and minimize the 1-affect-n phenomenon. HGKM is a tailored group key management approach to be deployed within the wireless cell to efficiently perform key management operations, while MOKM aims to tackle the

performance issue - multiple-membership change. These three proposed approaches each tackle specific issues in wireless group key management. In order to efficiently and securely perform group key management in the cellular wireless network, these three approaches need to be integrated to form a comprehensive wireless group key management solution. The object of this solution is to tackle the operational efficiency and security issues in group key management for the cellular wireless network. To better illustrate the operations of the proposed wireless group key management solution, we develop a case study to which the proposed solution is applied. In this case study, a real-world scenario is created in which a service provider offers three secure group applications in the cellular wireless network. The operations of the comprehensive solution are investigated based on four group operations: join, leave, handoff and membership switch (switch). In addition, in order to verify the efficiency and security of this integrated solution, topology matching key management (TMKM) [Sun & Trappe et al., 2002, 2003, 2004] is set as the benchmark. TMKM is chosen as the benchmark because it is currently the only available group key management system for the cellular wireless network.

The remainder of this chapter is organized as follows. Section 6.1 introduces the structure of the proposed comprehensive wireless group key management solution. In section 6.2, the operations of the proposed solution are investigated in detail by applying it to the case study. In section 6.3, through a comparison with TMKM, the comprehensive solution is analyzed and evaluated in terms of system performance and security. In section 6.4, the group key management system is formulized using statistics. This formulization helps system designers evaluate group key management

systems. Finally, section 6.5 summarizes the chapter.

6.1 A Comprehensive Group Key Management Solution for the Cellular Wireless Network

In the previous chapters, we have proposed three group key management approaches, each of them addressing a particular key management problem. However, in order to address the performance efficiency and security issues in the cellular wireless network from a systematic point of view, a comprehensive solution is required. Thus, in this chapter, we integrate these three proposed wireless group key management approaches to form a comprehensive solution to tackle efficiency and security problems in wireless group key management. The structure of this comprehensive solution is shown in Figure 6.1.

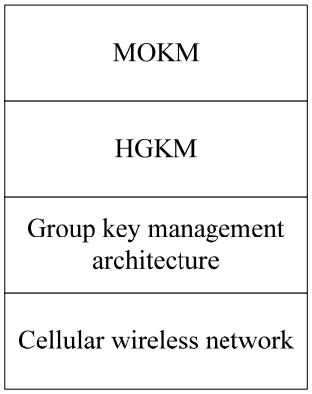


Figure 6.1 The comprehensive wireless group key management solution

From Figure 6.1, it can be observed that the comprehensive solution has a layered structure. The first layer (the bottom layer) is the cellular wireless network that provides not only the wireless transmission service, but also the network-specific

features for the upper layers. The second layer is the group key management architecture that seamlessly integrates the wireless key management infrastructure with the underlying wireless network topology to utilize the capacity of the wireless network to address scalability and the 1-affect-n issue. Based on the key management architecture, hybrid group key management (HGKM) is applied in the third level to organize group members within the wireless cell for the purpose of efficiently performing key management. Another group key management approach, membership-oriented key management (MOKM), is deployed in the topmost layer, providing a key-group management structure based on the memberships to tackle the multiple-membership change issue. Figure 6.2 illustrates the logical structure of the proposed comprehensive wireless group key management solution.

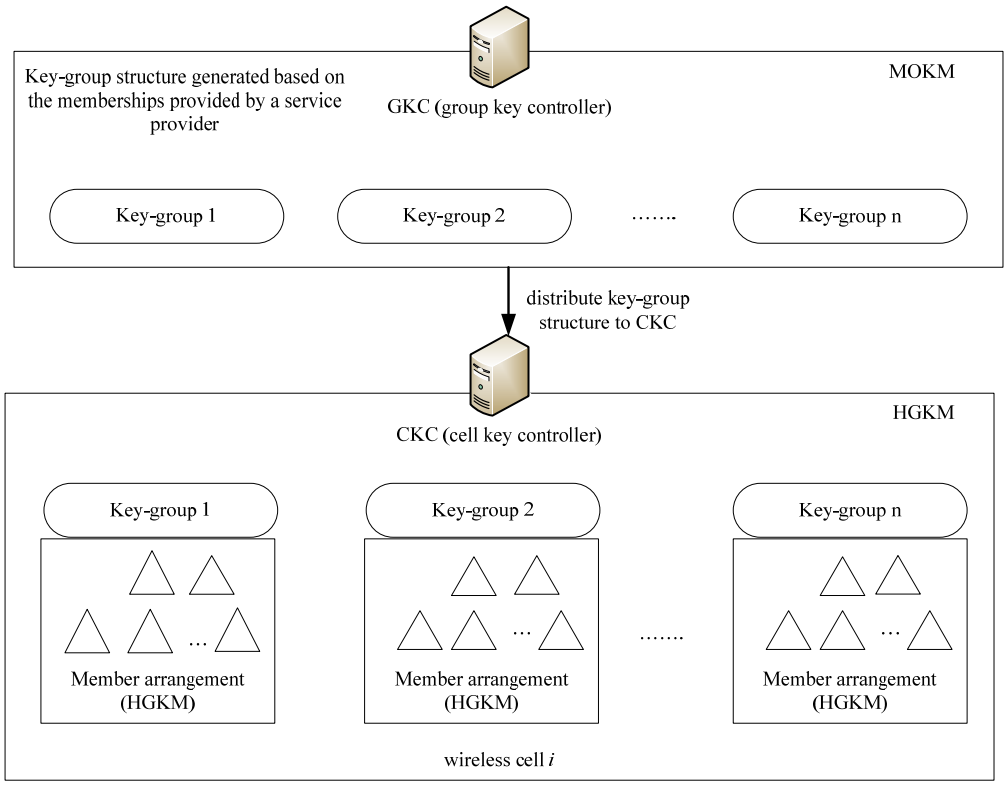


Figure 6.2 The key management structure of the proposed comprehensive solution

The group key controller (GKC) supplied by the group service provider generates a key-group structure based on the MOKM approach. This key-group structure is then distributed to all cell key controllers (CKCs) for the purpose of key management within the cellular wireless network. In a wireless cell, CKC applies HGKM to generate operation units for each key-group to perform key management operations.

The features of this comprehensive wireless group key management solution are:

- efficient key management on a large scale that minimizes the 1-affect-n phenomenon (provided by the proposed wireless group key management architecture (WGKMA));
- efficient key management within the wireless cell by applying micro-key management in each key-group (provided by HGKM); and
- efficient key management in the event of multiple-membership change (provided by MOKM).

In the following sections, we apply our solution to a case study. The creation of a real-world scenario allows the operations of this comprehensive wireless group key management solution to be discussed in detail.

6.2 The Case Study

6.2.1 Group Key Management Initiation

The multimedia company *M* has three multimedia TV channels dedicated to news, sports and movies. Each channel is associated with a global multicast IP address, 225.10.10.1, 225.10.10.2 and 225.10.10.3 for news, sports and movies respectively. Each channel is a secure group application, in which the communication content is encrypted by a group traffic encryption key (i.e. $GTEk_{news}, GTEK_{sports}, GTEK_{movies}$). Based on these three channels, service provider *M* offers several program packages to which users can subscribe. The program packages are defined as follows.

- Program package 1: news only
- Program package 2: sports only
- Program package 3: movies only
- Program package 4: news and sports
- Program package 5: sports and movies
- Program package 6: news, sports and movies

In the initialization stage, the GKC applies the MOKM approach to create a key-group structure according to the above six program packages. Figure 6.3 shows this key-group structure.

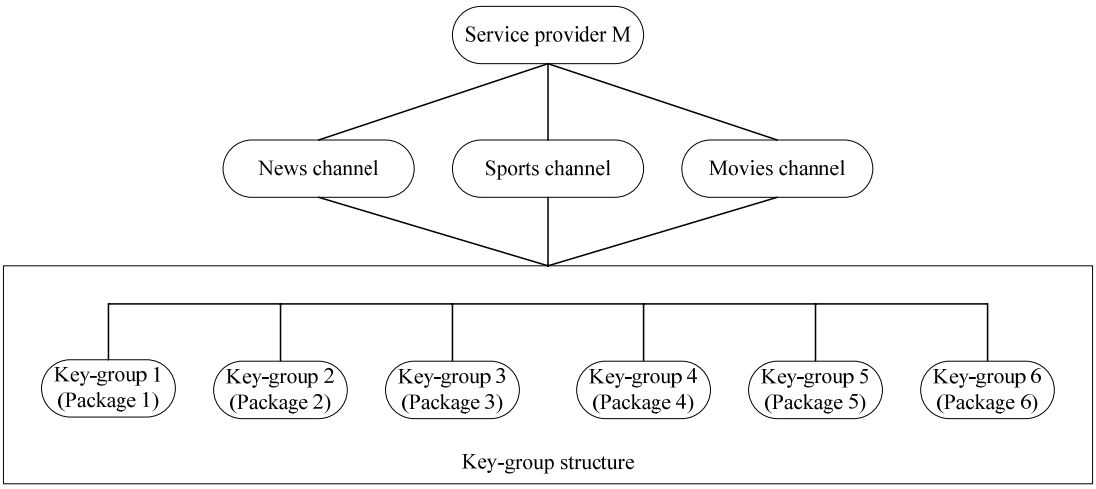


Figure 6.3 Key-group structure of service provider *M*

After generating this key-group structure, the GKC distributes the structure to all the CKCs within the cellular wireless network via multicast transmission. When a CKC receives the structure, the CKC initializes the group key management within the wireless cell in three steps based on the key-group's structure.

- Step 1: Key generation

  The CKC generates cell traffic encryption keys ($\mathrm{CTEK}_{\mathrm{news}}$, $\mathrm{CTEK}_{\mathrm{sports}}$ and $\mathrm{CTEK}_{\mathrm{movies}}$) for these three applications and the key-group keys, for each key group ($k_{\text{key-group\_1}}, k_{\text{key-group\_2}}, ..., k_{\text{key-group\_6}}$).

- Step 2: Assignment of local IP multicast addresses

  The CKC assigns local IP multicast addresses to service provider M ($\mathrm{IP}(M)$) and each key-group ($\mathrm{IP}(\text{key-group\_}n)$) for the purpose of key management. In order to fully use the local IP multicast addresses, the tuple ($\mathrm{IP}(M), \mathrm{IP}(\text{key-group\_}n)$) is applied to identify the key-group for the purpose of

key management at the key-group level.

- Step 3: Key-group initiation

  In this step, the CKC starts to build operation units within each key-group. The CKC assigns local IP multicast addresses (IP(operation_unit)) to each generated operation unit. The tuple (IP(key-group_n) , IP(operation_unit)) is applied to identify the operation unit for the purpose of key management at the operation unit level.

In this case study, we assume there are two users, Alice and Bob, in cell 1. Figure 6.4 illustrates the scenario. Without loss of generality, we apply a binary tree to each operation unit. In the following sections, we illustrate the operations of the proposed comprehensive solution from four member actions: join, membership switch (switch), handoff and leave.

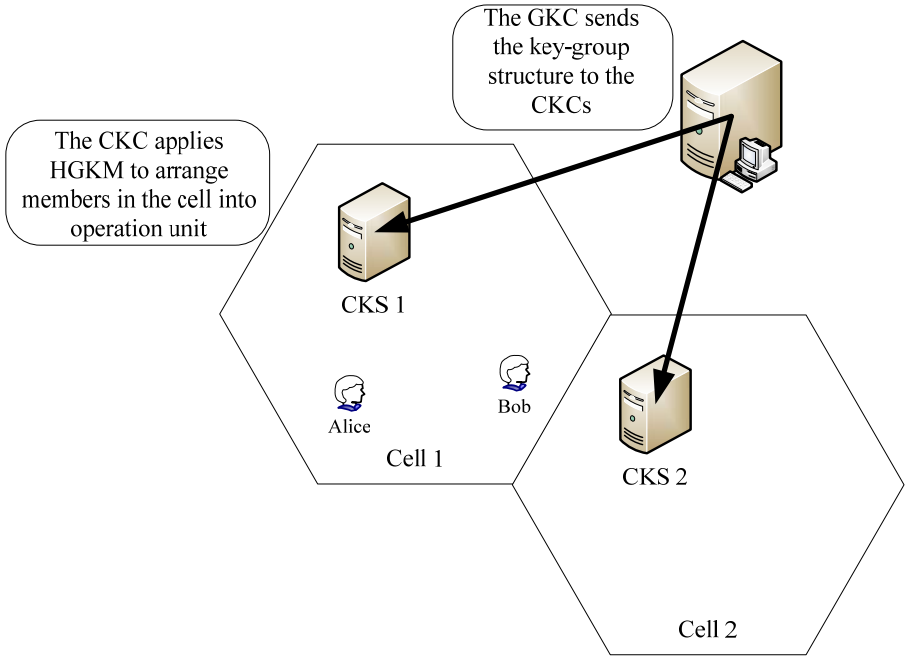


Figure 6.4 Scenario of the case study

6.2.2 The Join Operation

At time t_1, Alice joins service provider *M* and subscribes to program package 1, the news channel. There are two steps in Alice's join.

- Step 1: Service registration

In this step, Alice sends a join request to the GKC for service registration.

$$\text{Alice} \rightarrow \text{GKC} : \{\text{request for joining program package 1}\}$$

After successful authentication, the GKC sends Alice the group TEK for the news ($\text{GTEK}_{\text{news}}$) and notifies CKC_1 in the wireless cell 1 of Alice's join.

$$\text{GKC} \rightarrow \text{Alice} : \{\text{ATEK}_{\text{news}}\} k_{\text{GKC-Alice}}$$

- Step 2: Key-group join

In order to receive group data, Alice sends a request to CKC_1 via IGMP (Internet Group Management Protocol) for data receiving.

$$\text{Alice} \rightarrow \text{CKC\_1} : \{\text{request for data receiving}\}$$

CKC_1 searches for an available slot within key-group 1 for Alice after receiving the successful authentication of Alice from the GKC. CKC_1 finds a vacant node in member unit 1 (MU_1) and assigns Alice to it. We assume there are eight members in member unit 1 after Alice joins, as shown in Figure 6.5. In order to enforce backward secrecy, after Alice joins member unit 1, CKC_1 generates a new CTEK, $\text{CTEK}_{\text{news}}'$, for the news channel, and a new key-group key, $k_{\text{key-group\_1}}'$, for key-group 1. CKC_1 also generates the new supporting keys (k_{18}', k_{14}' and k_{34}') to replace the affected keys in member unit 1. After key generation, CKC_1 sends Alice

the set of keys Alice is entitled to know.

$$\text{CKC\_1} \rightarrow \text{Alice} : \{\text{CTEk}_{\text{news}}', k_{\text{key-group\_1}}', k_{18(\text{MU\_1})}', k_{14(\text{MU\_1})}', k_{34(\text{MU\_1})}'\} k_{4(\text{MU\_1})}$$

After joining the service, Alice has two TEKs, $\text{GTEk}_{\text{news}}$ and $\text{CTEK}_{\text{news}}$, for the news channel; one is a group TEK (GTEK) and the other is cell TEK (CTEK). Now, Alice can decrypt the program contents using these two keys.

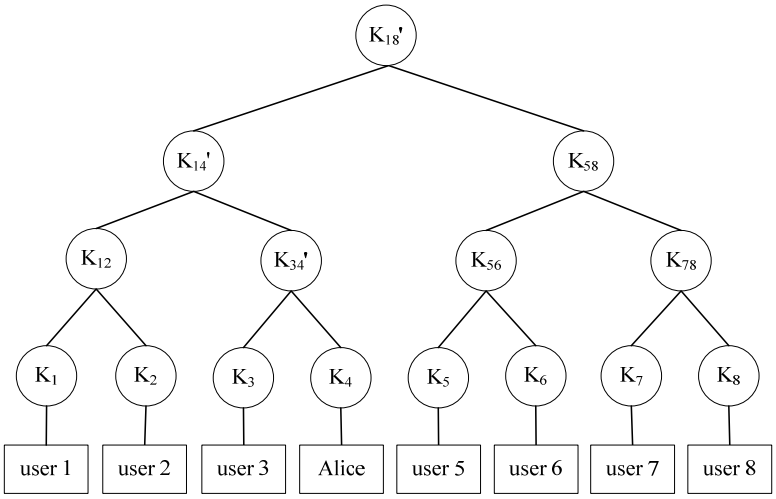


Figure 6.5 Member unit 1 (MU_1) in key-group 1 (after Alice's join)

After sending the keys to Alice, CKC_1 follows the rekeying procedure in HGKM to update keys for the remaining members in the key-group. Three steps are involved to fulfill rekeying.

First, CKC_1 updates keys for the members in member unit 1, the directly-affected operation unit. CKC_1 follows the rekeying process for joining member unit in HGKM as described in detail in section 4.2.3. CKC_1 creates an integrated rekeying message containing all rekeying information for the members in member unit 1. CKC_1 multicasts this integrated message to the directly-affected member unit 1 via the local IP address of member unit 1.

$$\text{CKC\_1} \Rightarrow \text{MU\_1: } \{\{k_{18}'\}k_{58}, \{k_{18}', k_{14}'\}k_{12}, \{k_{18}', k_{14}', k_{34}'\}k_3, \{k_{18}'\}k_{\text{CKC-unit\_leader}}\}$$

After receiving this rekeying message, the members in member unit 1 can gain the latest supporting keys from the corresponding section.

Second, after rekeying has been completed in the directly-affected member unit, CKC_1 needs to update $\text{CTEK}_{\text{news}}$ and the key-group key for the other operation units within key-group 1. CKC_1 multicasts a rekeying message, which contains the new keys encrypted by the current key-group key, to key-group 1.

$$\text{CKC\_1} \Rightarrow \text{key-group 1:} \{\text{CTEK}_{\text{news}}', k_{\text{key-group\_1}}'\}k_{\text{key-group\_1}}$$

Finally, following the rekeying procedure of MOKM, CKC_1 multicasts the new $\text{CTEK}_{\text{news}}$ to the other affected key-groups, key-groups 4 and 6, which also need to know the CTEK change for the news channel.

$$\text{CKC\_1} \Rightarrow \text{ key-group 4, 6: } \{\{\text{CTEK}_{\text{news}}'\}k_{\text{key-group\_4}}, \{\text{CTEK}_{\text{news}}'\}k_{\text{key-group\_6}}\}$$

When the remaining members in key-groups 4 and 6 receive this rekeying message, they can update their CTEKs for the news channel.

Based on HGKM and MOKM approach, we can observe that CKC_1 sends four rekeying messages for key updating during Alice's join. The total number of keys encrypted by CKC_1 for Alice's join (join member unit) is:

$$\frac{(h_{\text{unit}}+1)(h_{\text{unit}}+2)}{2} + 2 + n_{\text{CTEK}}(\text{affected}) + \sum_{i=1}^{n_{\text{CTEK}}(\text{affected})} i \times n_{\text{key-group}}(i)$$

where h_{unit} is the height of the key tree for the operation unit, $n_{\text{CTEK}}(\text{affected})$ is the number of affected CTEKs and $n_{\text{key-group}}(i)$ is the number of key-groups affected by i new CTEK(s).

Now we consider Bob's join. At time t_2, Bob decides to join the service and subscribes to program package 5 to receive the sports and movies. Similar to Alice, Bob sends a request to the GKC to join program package 5. The GKC authenticates Bob and sends Bob two group TEKs ($GTEK_{sports}$ and $GTEK_{movies}$) for sports and movie channel respectively.

$$GKC \rightarrow Bob : \{GTEK_{sports}, GTEK_{moives}\}k_{GKC\text{-}Bob}$$

In cell 1, CKC_1 assigns Bob to leader unit 2 (LU_2) in key-group 5 (shown in Figure 6.6).

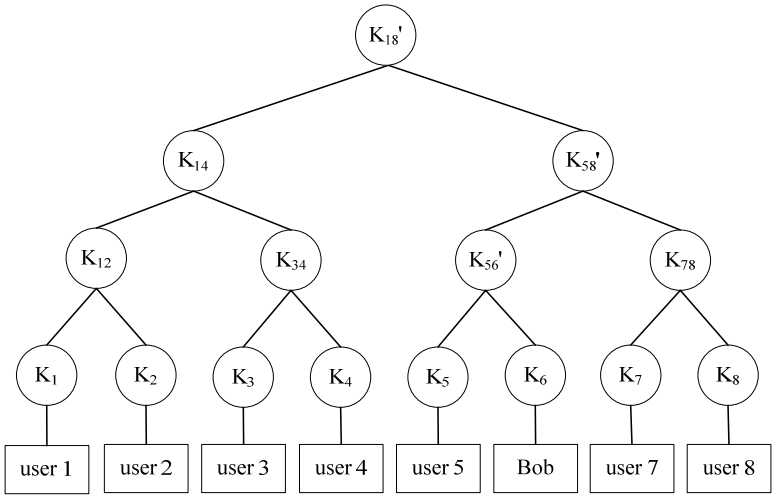


Figure 6.6 Leader unit 2 (LU_2) in key-group 5 (after Bob's join)

CKC_1 generates the following new keys for rekeying:

- new CTEKs, $CTEK_{sports}'$ and $CTEK_{movies}'$, for the sports and movie channels;
- new key-group key, $k_{key\text{-}group\_5}'$, for key-group 5; and
- new supporting keys, $k_{18(LU\_2)}', k_{58(LU\_2)}'$ and $k_{56(LU\_2)}'$, for leader unit 2.

Then, CKC_1 sends the new keys to Bob.

$$\text{CKC\_1} \rightarrow \text{Bob}: \{\text{CTEK}_{\text{sports}}', \text{CTEK}_{\text{movies}}', k_{\text{key-group\_5}}', k_{18(\text{LU\_2})}', k_{58(\text{LU\_2})}', k_{56(\text{LU\_2})}'\} k_{6(\text{LU\_2})}$$

After receiving this rekeying message, Bob has four TEKs, ($\text{GTEK}_{\text{sports}}$, $\text{CTEK}_{\text{sports}}$, $\text{GTEK}_{\text{movies}}$ and $\text{CTEK}_{\text{movies}}$), to decrypt the group communication for the sports and movie channels.

After rekeying Bob, following the rekeying procedure in HGKM and MOKM, CKC_1 updates keys for the members in leader unit 2, other operation units in key-group 5 and other key-groups affected by the change of CTEKs in turn. The rekeying procedure is similar to that of Alice's join.

$$\text{CKC\_1} \Rightarrow \text{LU\_2}: \{\{k_{18}'\} k_{14}, \{k_{18}', k_{58}'\} k_{78}, \{k_{18}', k_{58}', k_{56}'\} k_5\}$$

$$\text{CKC\_1} \Rightarrow \text{key-group 5}: \{\text{CTEK}_{\text{sports}}', \text{CTEK}_{\text{movies}}', k_{\text{key-group\_5}}'\} k_{\text{key-group\_5}}$$

$$\begin{aligned}\text{CKC\_1} \Rightarrow \text{ key-group 2, 3, 4, 6}: \{\{&\text{CTEK}_{\text{sports}}'\} k_{\text{key-group\_2}}, \{\text{CTEK}_{\text{movies}}'\} k_{\text{key-group\_3}}, \\ &\{\text{CTEK}_{\text{sports}}'\} k_{\text{key-group\_4}}, \\ &\{\text{CTEK}_{\text{sports}}', \text{CTEK}_{\text{movies}}' k_{\text{key-group\_6}}\}\}\end{aligned}$$

In line with the HGKM and MOKM approach, during the rekeying procedure for Bob's join, CKC_1 sends four rekeying messages for key updating (leader unit join). The total number of keys encrypted by CKC_1 during the rekeying is:

$$\frac{(h_{\text{unit}}+1)(h_{\text{unit}}+2)}{2} + 1 + n_{\text{CTEK}}(\text{affected}) + \sum_{i=1}^{n_{\text{CTEK}}(\text{affected})} i \times n_{\text{key-group}}(i)$$

where h_{unit} is the height of the key tree for the operation unit, $n_{\text{CTEK}}(\text{affected})$ and $n_{\text{key-group}}(i)$ are the number of affected CTEKs and key-groups affected by i new CTEK(s).

Table 6.1 summarizes the operational costs of join for Alice's and Bob's join.

Table 6.1 The communication and computation cost for Alice's and Bob's join

	Communication cost	Computation cost
Alice (member unit join)	4	$\frac{(h_{\text{unit}}+1)(h_{\text{unit}}+2)}{2}+2+n_{\text{CTEK}}(\text{affected})+\sum_{i=1}^{n_{\text{CTEK}}^{(\text{affected})}} i\times n_{\text{key-group}}(i)$
Bob (leader unit join)	4	$\frac{(h_{\text{unit}}+1)(h_{\text{unit}}+2)}{2}+1+n_{\text{CTEK}}(\text{affected})+\sum_{i=1}^{n_{\text{CTEK}}^{(\text{affected})}} i\times n_{\text{key-group}}(i)$

h_{unit}: the height of the key tree for the operation unit

$n_{\text{CTEK}}(\text{affected})$: the number of affected CTEKs

$n_{\text{key-group}}(i)$: the number of key-groups in MOKM, which are affected by i new CTEK(s)

6.2.3 The Membership Switch Operation

Supposing at time t_3, Bob wants to change his plan from program package 5 to program package 3, (Bob has decided to leave the sports channel). Bob sends the package change request to the GKC. After authentication, the GKC approves Bob's request. Due to the presence of MOKM in the solution, the GKC does not need to immediately update the group TEK, $\text{GTEK}_{\text{sports}}$, which is affected by Bob's switch. The GKC simply notifies CKC_1 of Bob's membership switch.

$$\text{GKC} \rightarrow \text{CKC\_1}: \{\text{Bob changes to program package 3}\}$$

In order to ensure forward secrecy, CKC_1 updates its cell TEK for the sports channel immediately. CKC_1 generates a new CTEK, $\text{CTEK}_{\text{sports}}'$, for the sports channel, and sends this new CTEK to the affected key-groups that need to know the

change of this CTEK. The new CTEK is encrypted by the key-group keys of affected key-groups.

$$\text{CKC\_1} \Rightarrow \text{ key-groups 2, 4, 6}: \{\{\text{CTEK}_{\text{sports}}'\}k_{\text{key-group\_2}}\}, \{\text{CTEK}_{\text{sports}}'\}k_{\text{key-group\_4}}\}, \{\text{CTEK}_{\text{sports}}'\}k_{\text{key-group\_6}}\}\}$$

After receiving this rekeying message, the members in key-groups 2, 4 and 6 use their key-group keys to decrypt the message to obtain the latest CTEK for the sports channel.

Due to the switch operation, CKC_1 reassigns Bob to key-group 3. Bob leaves key-group 5 and joins key-group 3. CKC_1 invokes the rekeying process of leave to update the supporting keys for leader unit 2 in key-group 5 that Bob leaves. The CKC generates the new supporting keys and sends them to leader unit 2.

$$\text{CKC\_1} \rightarrow \{\text{leader unit 2}\}: \{\text{new supporting keys for key updating}\}$$

After this, in line with the rekeying procedure in HGKM, CKC_1 needs to update the CTEK and the key-group key for the remaining members in key-group 5. CKC_1 sends the new CTEK, $\text{CTEK}_{\text{sports}}'$ and the new key-group key, $k_{\text{key-group\_5}}'$, to the leader units in key-group 5.

$$\text{CKC\_1} \Rightarrow \text{leader-unit-group in key-group 5: } \{\{\text{CTEK}_{\text{sports}}', k_{\text{key-group\_5}}'\}k_{\text{Leader\_unit\_1}}, \cdots, \{\text{CTEK}_{\text{sports}}', k_{\text{key-group\_5}}'\}k_{\text{Leader\_unit\_n}}\}$$

When they receive these new keys, the leaders in key-group 5 distribute these two new keys within their own member units.

$$\text{leader 1} \Rightarrow \text{member unit 1: } \{\{\text{CTEK}_{\text{sports}}', k_{\text{key-group\_5}}'\}k_{\text{member\_unit\_1}}$$

.....

$$\text{leader n} \Rightarrow \text{member unit n: } \{\{\text{CTEK}_{\text{sports}}', k_{\text{key-group\_5}}'\}k_{\text{member\_unit\_n}}$$

After rekeying key-group 5, CKC_1 reassigns Bob to member unit 6 (MU_6) in key-group 3 and sends the new generated supporting keys to Bob.

$$\text{CKC\_1} \rightarrow \text{Bob} : \{\text{new generated supporting keys in MU\_6}\}k_{\text{CKC\_1-Bob}}$$

If Bob joins the movie channel earlier than the latest rekeying within member unit 6, there is no need to perform the rekeying, because at the rekeying time, Bob is already a member of the movie channel. Otherwise, in order to enforce strict backward secrecy, CKC_1 needs to perform key updating following the rekeying procedure of joining member unit that is the same as that of Alice's join.

In conclusion, due to the MOKM's presence, the membership switch operation only affects two key-groups. The switching member leaves the current key-group and joins a newly-assigned key-group based on its latest membership(s). The operational cost can therefore be treated as a summation of one leave and one join operation.

6.2.4 The Handoff Operation

In terms of handoff, supposing at time t_4, Alice moves from cell 1 to cell 2. During the handoff, Alice follows the handoff procedure in HGKM (discussed in section 4.2.5) to send a handoff join request to CKC_2. When CKC_2 receives this request, CKC_2 contacts CKC_1 to gain authentication for Alice.

$$\text{CKC\_2} \rightarrow \text{CKC\_1} : \{\text{Is Alice is an authenticated group member?}\}$$
$$\text{CKC\_1} \rightarrow \text{CKC\_2} : \{\text{Yes, she is.}\}$$

After successful authentication, CKC_2 assigns Alice to leader unit 3 (LU_3) as a leader candidate in key-group 1. CKC_2 sends Alice the cell TEK ($\text{CTEK}_{\text{news}}$) and the supporting keys Alice is entitled to know, as shown in Figure 6.7

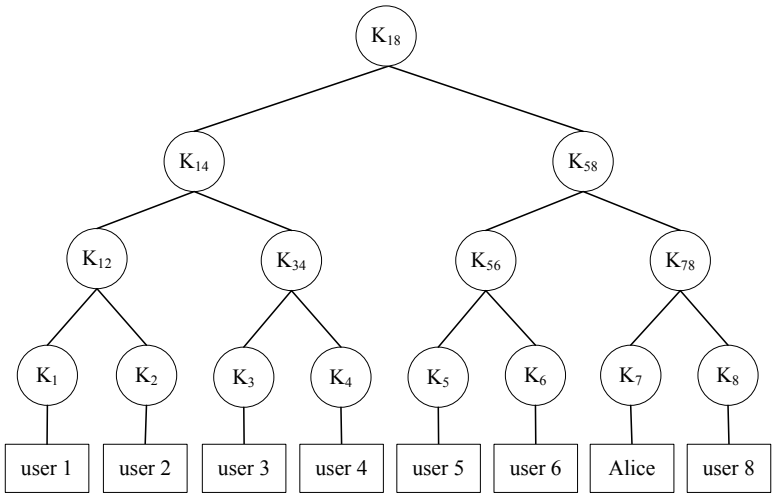


Figure 6.7 Leader unit 3 (LU_3) in cell 2 (after Alice's handoff)

$$\text{CKC\_2} \rightarrow \text{Alice} : \{\text{CTEK}_{\text{news}}(\text{cell\_2}), k_{\text{key-group\_1}}, k_{18(\text{LU\_3})}, k_{58(\text{LU\_3})}, k_{78(\text{LU\_3})}\} k_{7(\text{LU\_3})}$$

After Alice finishes handoff, CKC_2 sends a handoff confirmation message to CKC_1 to inform it of the completion of Alice's handoff.

$$\text{CKC\_2} \rightarrow \text{CKC\_1} : \{\text{Alice's handoff completes}\}$$

Once CKC_1 receives this confirmation, CKC_1 places Alice in the post-handoff-user list and starts a timer to keep Alice in this list for a set period. If Alice moves back to cell 1 before the time has expired, CKC_1 is able to find Alice in the list and there is no need to perform any rekeying. Alice can still use all the CTEKs and supporting keys she has stored for cell 1 before her handoff. Otherwise, after the time has expired, CKC_1 marks the previous node occupied by Alice as being now available for future assignment.

6.2.5 The Leave Operation

Finally, we investigate the leave operation in this case study. Supposing at time t_5 Alice wishes to leave the group. She sends a leave message to the GKC and then the GKC sends out a message to all CKCs to inform them of the departure of Alice. CKC checks its user list and handoff list to decide whether it needs to perform rekeying within the cell.

$$\text{Alice} \rightarrow \text{GKC: \{leaves group application\}}$$
$$\text{GKC} \Rightarrow \text{CKCs: \{Alice leaves the group application\}}$$

Because Alice is in the leader unit, based on HGKM (described in section 4.2.4), three scenarios need to be considered for the leave operation:

- a leadership candidate leaves the group;
- a leader leaves the group and a leadership candidate is available to be the new leader; and
- a leader leaves the group and no leadership candidate is available.

In this case study, Alice is a leadership candidate. When CKC_2 is notified of Alice's leave, CKC_2 generates a new CTEK ($CTEK_{news}'$) for the news channel and invokes the rekeying process to update affected keying materials. Starting at the bottom of the hierarchy, three steps fulfill this rekeying.

- Step 1: Key updating in the directly-affected operation unit

In this step, CKC_2 updates the supporting keys for leader unit 3 in key-group 1 that Alice leaves. Figure 6.8 shows the key management structure of leader unit 3 after Alice leaves.

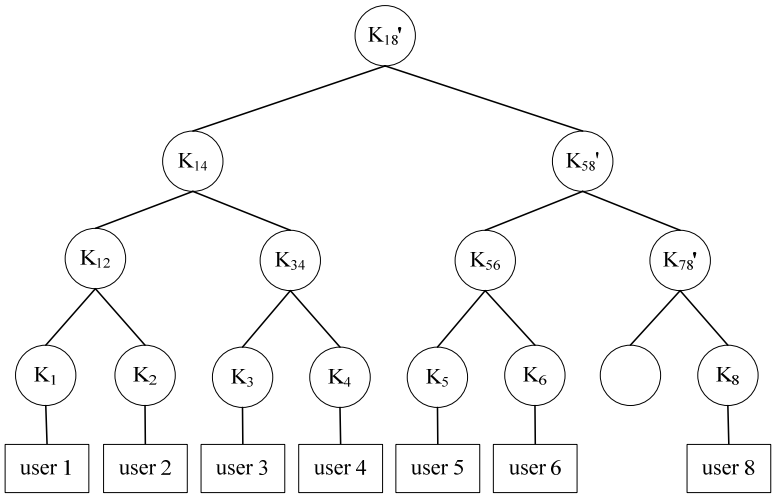


Figure 6.8 Leader unit 3 (LU_3) in key-group 1 (after Alice's leave)

Following the rekeying procedure for leave in HGKM (described in section 4.2.4), CKC_2 sends the following rekeying messages to the members in leader unit 3 for the key updating.

CKC_2 $\rightarrow$ leader unit 3: {for user 1-4: $\{k_{18(LU\_3)}'\}k_{14(LU\_3)}$

for user 5,6 : $\{k_{18(LU\_3)}', k_{58(LU\_3)}'\}k_{56(LU\text{-}3)}$

for user 8 : $\{k_{18(LU\_3)}', k_{58(LU\_3)}', k_{78(LU\_3)}'\}k_{8(LU\_3)}\}$

- Step 2: key updating in the directly-affected key-group

After rekeying leader unit 3, CKC_2 needs to update the $CTEK_{news}$ for all the remaining members in key-group 1. CKC_2 distributes the new $CTEK_{news}$ within the leader-unit-group to update the CTEK for the leaders. The leaders then distribute the new keys within their own member units.

CKC_2 $\Rightarrow$ leader-unit-group (key-group 1): $\{\{CTEK_{news}', k_{key\text{-}group\_1}'\}k_{Leader\_unit\_1}, \cdots\cdots,$

$\{CTEK_{news}', k_{key\text{-}group\_1}'\}k_{Leader\_unit\_n}\}$

leader 1 $\Rightarrow$ member unit 1: $\{\{CTEK_{news}', k_{key\text{-}group\_1}'\}k_{member\_unit\_1}$

.....

leader n $\Rightarrow$ member unit n: $\{\{CTEK_{news}', k_{key\text{-}group\_1}'\}k_{member\_unit\_n}$

- Step 3: key updating for other affected key-groups

Key groups 4 and 6 are also affected by the leave of Alice, because the CTEK for the news channel has changed. Therefore, CKC_2 multicasts this new CTEK to all other affected key-groups to update their CTEK.

CKC_2 $\Rightarrow$ key-groups 4 and 6: $\{\{\text{CTEK}_{\text{news}}'\}k_{\text{key-group\_4}}, \{\text{CTEK}_{\text{news}}'\}k_{\text{key-group\_6}}\}$

In using the HGKM and MOKM approach, CKC_2 sends three rekeying messages for key updating after Alice leaves the service. The total number of keys encrypted by CKC_2 during the leave rekeying is:

$$\frac{(h_{\text{unit}}+1)\times h_{\text{unit}}}{2} + (n_{\text{CTEK}}(\text{affected})+1)\times n_{\text{leader\_unit}} + \sum_{i=1}^{n_{\text{CTEK}}(\text{affected})} i \times n_{\text{key-group}}(i)$$

where h_{unit} is the height of the key tree for the operation unit, $n_{\text{CTEK}}(\text{affected})$ and $n_{\text{key-group}}(i)$ are the number of affected CTEKs and the number of key-groups affected by i new CTEK(s), and $n_{\text{leader\_unit}}$ is the number of leader units in the key-group.

In terms of the other two leave scenarios, the rekeying procedure is similar to Alice's leave. Based on the formulas in HGKM and MOKM, we can easily calculate their communication and computation costs, presented in Table 6.2.

Now we consider Bob's leave. Because Bob is in a member unit, the key updating follows the process of the member unit leave operation (investigated in detail in section 4.2.4). Figure 6.9 illustrates the key tree in member unit 6 after Bob leaves.

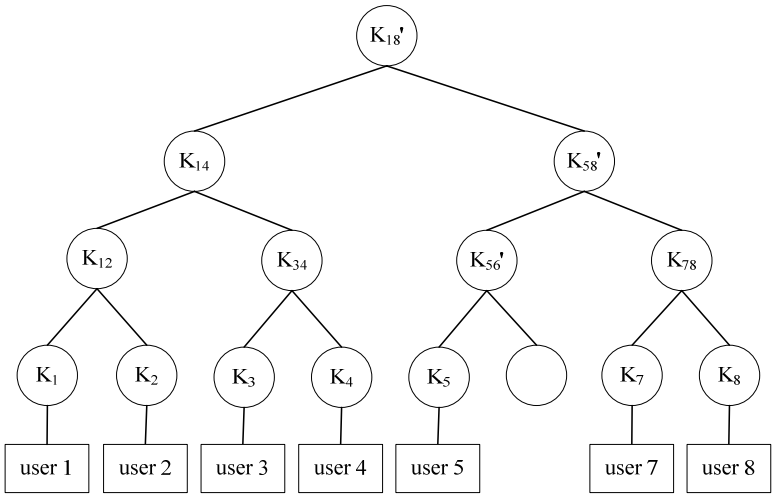


Figure 6.9 Member unit 6 (MU_6) in key-group 3 (after Bob's leave)

Bob subscribes only to the movie channel before he leaves the service. Therefore, after Bob's leave, CKC_1 generates a new CTEK ($CTEK_{movies}'$) for the movie channel. Similar to the rekeying for Alice's leave, Bob's leave also requires three steps for key updating.

First, CKC_1 updates keys for the members in member unit 6 that Bob leaves. CKC_1 generates new supporting keys for member unit 6 and sends these new keys to the affected members.

$$\begin{aligned} \text{CKC\_1} \Rightarrow \text{member unit 6: } \{&\text{for user 1- 4: } \{k_{18}'\}k_{14}, \\ &\text{for user 7,8: } \{k_{18}', k_{58}'\}k_{78}, \\ &\text{for user 5: } \{k_{18}', k_{58}', k_{56}'\}k_5, \\ &\text{for the leader of MU\_6: } \{k_{18}'\}k_{\text{CKC\_unit-leader}}\} \end{aligned}$$

Second, CKC_1 updates the $CTEK_{news}$ for the other operation units in key-group 3. CKC_1 multicasts the new CTEK, $CTEK_{news}'$, to the leader-unit-group. After receiving this new CTEK, the leaders distribute the latest CTEK within their

member units.

$$\text{CKC\_1} \Rightarrow \text{leader-unit-group (key-group 3): } \{\{\text{CTEK}_{\text{movies}}', k_{\text{key-group\_3}}'\}k_{\text{Leader\_unit\_1}}, \ldots\ldots, \{\text{CTEK}_{\text{movies}}', k_{\text{key-group\_3}}'\}k_{\text{Leader\_unit\_n}}\}$$

$$\text{leader 1} \Rightarrow \text{member unit 1: } \{\{\text{CTEK}_{\text{movies}}', k_{\text{key-group\_3}}'\}k_{\text{member\_unit\_1}}$$

.....

$$\text{leader n} \Rightarrow \text{member unit n: } \{\{\text{CTEK}_{\text{movies}}', k_{\text{key-group\_3}}'\}k_{\text{member\_unit\_n}}$$

Finally, CKC_1 updates CTEK for the other affected key-groups.

$$\text{CKC\_1} \Rightarrow \text{key-groups 5, 6: } \{\{\text{CTEK}_{\text{movies}}'\}k_{\text{key-group\_5}}, \{\text{CTEK}_{\text{movies}}'\}k_{\text{key-group\_6}}\}$$

From the rekeying process for Bob's leave, it can be observed that CKC_1 sends three rekeying messages during the rekeying. The total number of keys encrypted by CKC_1 is:

$$\frac{(h_{\text{unit}}+1)\times h_{\text{unit}}}{2} + 1 + (n_{\text{CTEK}}(\text{affected})+1)\times n_{\text{leader\_unit}} + \sum_{i=1}^{n_{\text{CTEK}}(\text{affected})} i \times n_{\text{key-group}}(i)$$

where h_{unit} is the height of the operation unit, $n_{\text{CTEK}}(\text{affected})$ is the number of affected CTEKs, $n_{\text{key-group}}(i)$ is the number of key-groups affected by i new CTEK(s), and $n_{\text{leader\_unit}}$ is the number of leader units in the key-group.

Table 6.2 summarizes the communication and computation cost for Alice's and Bob's leave.

Table 6.2 The communication and computation costs for Alice's and Bob's leave

	Communication cost	Computation cost
Bob's leave (member leaves member unit)	3	$\frac{(h_{\text{unit}}+1)\times h_{\text{unit}}}{2}+1+(n_{\text{CTEK}}(\text{affected})+1)\times n_{\text{leader\_unit}}+\sum_{i=1}^{n_{\text{CTEK}}(\text{affected})} i\times n_{\text{key-group}}(i)$
Alice's leave (leadership candidate leaves)	3	$\frac{(h_{\text{unit}}+1)\times h_{\text{unit}}}{2}+(n_{\text{CTEK}}(\text{affected})+1)\times n_{\text{leader\_unit}}+\sum_{i=1}^{n_{\text{CTEK}}(\text{affected})} i\times n_{\text{key-group}}(i)$
A leader leaves the group and leadership candidate is available to be new leader	4	$\frac{(h_{\text{unit}}+1)\times h_{\text{unit}}}{2}+5+(n_{\text{CTEK}}(\text{affected})+1)\times n_{\text{leader\_unit}}+\sum_{i=1}^{n_{\text{CTEK}}(\text{affected})} i\times n_{\text{key-group}}(i)$
A leader leaves the group and no leadership candidate is available	4	$\frac{(h_{\text{unit}}+1)\times h_{\text{unit}}}{2}+2+(n_{\text{CTEK}}(\text{affected})+1)\times n_{\text{leader\_unit}}+\sum_{i=1}^{n_{\text{CTEK}}(\text{affected})} i\times n_{\text{key-group}}(i)$

h_{unit} : the height of key tree for operation unit

$n_{\text{CTEK}}(\text{affected})$: the number of affected CTEKs

$n_{\text{key-group}}(i)$: the number of key-groups affected by i new CTEK(s)

$n_{\text{leader\_unit}}$: the number of leader units in the key-group

6.3 System Evaluation

In this section, we analyze and evaluate the proposed comprehensive wireless group key management solution for the cellular wireless network (CWGKM). This solution is evaluated using the performance and security assessment parameters discussed in section 3.2.2. In addition, we compare the proposed solution with topology matching key management (TMKM) [Sun & Trappe et al., 2002, 2003], described in section 2.6.1, to demonstrate the efficiency and security of the proposed solution. TMKM is chosen as the benchmark because it is the only currently available group key management system for the cellular wireless network.

6.3.1 Scalability

Scalability is an important issue in wireless group key management. A wireless group key management system cannot be considered a practical, deployable solution if it cannot achieve scalability. Scalability can be evaluated from three perspectives: the system's ability to cover a large geographic area; the system's ability to support various group sizes and the system's ability to handle dynamic group membership changes. The following section evaluates the scalability of CWGKM from these three perspectives.

- Ability to cover a large geographic area

In CWGKM, the group key management structure can be seamlessly integrated with the underlying cellular wireless network. Each cellular wireless cell is also a group key management area as well controlled by a CKC. As a result of the overlap

between the wireless cell and the group key management area, the group key management domain can be expanded with the extension of the cellular wireless network, thereby providing the ability to cover a large geographic area.

- Ability to support various group sizes

CWGKM applies decentralized infrastructure where each wireless cell is an independent key management area. Within the cell, each CKC independently performs group key management for group members while it receives control messages from the GKC of the service provider. This design can divide a large group into a number of small and manageable units, each of them controlled by a CKC. The design enables the proposed CWGKM to deal with various group sizes, especially large groups.

- Ability to handle dynamic group membership changes

Due to the decentralized structure in CWGKM, each CKC only needs to process the key management within its cell. Group membership changes can therefore be processed locally and in parallel by the CKCs. This gives the CWGKM the capacity to handle highly dynamic membership changes to avoid overwhelming the capacity of the GKC.

On the other hand, TMKM can also provide scalability for group key management because wireless base stations are involved in key management in TMKM. This involvement enables TMKM to extend the key management area to the whole cellular wireless domain. The base stations can decide whether they need to distribute the keying materials in their cells. This enables TMKM to adjust itself to fit various group sizes. However, compared to the CWGKM, TMKM has two

disadvantages: a complicated management structure and centralized key management.

- A complicated management structure

In TMKM, the GKC, base stations, group members and other controllers form a multi-layered hierarchical management structure (discussed in section 2.6.1). This key management structure complicates group key management, because a multiple-layer structure increases the number of levels involved in key management. In contrast, in CWGKM, the GKC, CKCs and group members form a two-tier key management structure to simplify key management in the cellular wireless network (described in detail in section 3.2.1).

- Centralized key management

TMKM involves a centralized structure, in which the GKC controls all group key management. This centralized key management carries the risk of single-point failure. If the GKC is out of service, the whole group key management system becomes unavailable. In addition, the capacity of the GKC may be overwhelmed by a large number of rekeying requests for membership changes. In contrast, in CWGKM, a decentralized key management structure is applied. This fully utilizes the capability of the underlying cellular wireless network to avoid single-point failure and provides flexibility in group key management.

In summary, both CWGKM and TMKM can provide scalability to cover a large geographic area and to handle various group sizes, because both these designs cooperate with the cellular wireless network to perform key management. However,

Due to its decentralized structure, CWGKM can use parallel key management to tackle the highly dynamic membership changes that remain a performance hurdle for TMKM. Moreover, TMKM faces the potential risk of single-point failure.

6.3.2 The 1-affect-n Phenomenon

As discussed in section 2.3, the 1-affect-n phenomenon is a serious performance issue for wireless group key management because it can reduce the quality of service of secure group applications. Due to the application of only one TEK in secure group applications, key updating resulting from membership changes affects all remaining members. In a large and highly distributed group, a member is unable to participate in future group communication if it cannot receive the latest TEK in time. The cure for the 1-affect-n phenomenon is therefore to apply multiple TEKs in group applications to minimize the impact of membership changes.

In CWGKM, based on the proposed wireless group key management architecture, multiple TEKs are applied to address the 1-affect-n problem. The GKC controls the GTEK used by the sender(s) to encrypt the communication data at the application level. Each CKC generates its own CTEK for the purpose of distribution of communication data. When the encrypted group data reaches the cell, the base station encrypts the content again by the CTEK before forwarding it within the cell. Once a member leaves the group, only the CTEK needs to be updated immediately and the GTEK can be kept intact because the rekeying procedure is restricted within the scope of the wireless cell. The group members outside the directly-affected cell are not affected by this rekeying. This double encryption can minimize the impact of

the 1-affect-n phenomenon. Moreover, as the wireless cell is a small area, the time to distribute the new key can be reduced and the members within the cell can receive the latest key more expeditiously.

In contrast to CWGKM, TMKM applies a single TEK. When a member leaves the group, the new key is generated by the GKC and distributed through the hierarchical management structure to reach every remaining member within the cellular wireless network. This process increases the duration of key distribution and the 1-affect-n problem for TMKM.

6.3.3 Performance Evaluation

Operational efficiency is the issue of highest priority when evaluating a wireless group key management system. A group key management system can be recognized as a practical solution only if it can satisfy the requirements of operational efficiency. Operational efficiency can be assessed from three perspectives: communication cost, computation cost and key storage cost. In CWGKM, the proposed HGKM and MOKM approaches are integrated together to address the performance issue. In contrast, TMKM applies a centralized ALX tree derived from the normal hierarchical key tree to manage the keying materials. Table 6.3 summarizes the operational costs of the above three parameters for CWGKM and TMKM. These formulas are obtained from Chapters 4 and 5, where HGKM and MOKM have been analyzed in depth. From Table 6.3, it can be observed that the operational costs for CWGKM are mainly proportional to the height of the key tree for the operation unit, that is, the size of the operation unit. In contrast, the costs for TMKM are determined by three

factors: the degree (α) and height (L) of the key tree and the number of users (x) attached to a certain node in level L. The analysis and evaluation in Chapters 4 and 5 have demonstrated that micro-key management in the operation unit of HGKM has better performance than that of a centralized hierarchical key tree. Therefore, we can infer that CWGKM can achieve better performance than TMKM. Moreover, in terms of the multiple-membership issue, the proposed CWGKM has an advantage over TMKM because CWGKM is a membership-oriented approach. In CWGKM, only one or two key-groups are affected by multiple-membership changes. However, TMKM is an application-oriented scheme, and each separate key tree is associated with a group application. Therefore, in TMKM, the costs of key management in the event of multiple-membership changes is a summation of several join and leave rekeying processes.

Based on the earlier analysis and evaluation of HGKM and MOKM, we conclude that the integration of these two group key management approaches provides CWGKM with the following two features.

- CWGKM performs micro-key management in the operation unit to achieve operational efficiency in communication, computation and key storage during the rekeying process.
- CWGKM uses the membership-oriented approach to efficiently tackle multiple-membership changes.

Table 6.3 The operational costs for CWGKM and TMKM

		CWGKM	TMKM
Communication (CKC)	Join	4	$\sum_{i=1}^{j}(L+1)$
	Leave	$3+\frac{1}{s_{\text{operation\_unit}}}$	$\sum_{i=1}^{l}(\alpha-1)(L-1)+(x_i-1)$
	Switch	$\text{Cost}_{\text{communicstion}}(\text{join})+\text{Cost}_{\text{communicstion}}(\text{leaving})$	$\sum\text{Cost}_{\text{communicstion}}(\text{join})+\sum\text{Cost}_{\text{communicstion}}(\text{leaving})$
Computation	Join	$(X+1)\times p_{\text{member\_unit\_join}})+(X+2)\times(1-p_{\text{member\_unit\_join}})$	$\sum_{i=1}^{j}\left(\frac{(L+1)(L+2)}{2}-1\right)_i$
	Leaving	Case I: $(Y+1)\times p_1(\text{I})+Y\times p_2(\text{I})+(Y+5)\times(1-p_1(\text{I})-p_2(\text{I}))$	$\sum_{i=1}^{l}\left((\alpha-1)(L-1)+(x_i-1)L\right)$
		Case II: $(Y+1)\times p_1(\text{II})+(Y+2)\times(1-p_1(\text{II}))$	
	Switch	$\text{Cost}_{\text{computation}}(\text{join})+\text{Cost}_{\text{computation}}(\text{leaving})$	$\sum_{i=1}^{j}\text{Cost}_{\text{computation}}(\text{join})+\sum_{i=1}^{l}\text{Cost}_{\text{computation}}(\text{leaving})$
Key storage	CKC	$\sum_{i=1}^{n_{\text{key-groups}}}(n_{\text{unit}}\times(2s_{\text{operation\_unit}}-1))+n_{\text{applications}}$	$\sum_{i=1}^{\text{n}_{\text{applications}}}(\frac{\alpha n_i-1}{\alpha-1}+\sum_{i}^{n}x_i)$

	Member (leader)	$h_{\text{unit}} + 2j + 2$	$\sum_{i=1}^{j} (L+1)_i$
	Member and leader candidate	$h_{\text{unit}} + 2j + 1$	

$s_{\text{operation\_unit}}$: the size of operation unit in HGKM.

$n_{\text{applications}}$: the number of group applications.

$n_{\text{member\_in\_member\_units}}$: the number of members in member units in CWGKM.

$n_{\text{total\_group\_members}}$: the total number of group members in CWGKM.

$n_{\text{leader\_candidate}}$: the number of leader candidates in CWGKM.

n_{unit} : the number of operation units in CWGKM.

$n_{\text{leader\_units}}$: the number of leader units in CWGKM

X : $\frac{(h_{\text{unit}}+1)(h_{\text{unit}}+2)}{2} + \sum_{i=1}^{n_{\text{CTEK}}(\text{affected})} (i \times n_{\text{key-group}}(i))$

Y : $\frac{(h_{\text{unit}}+1) \times h_{\text{unit}}}{2} + (n_{\text{CTEK}}(\text{affected})+1) \times n_{\text{leader\_units}} + \sum_{i=1}^{n_{\text{CTEK}}(\text{affected})} (i \times n_{\text{key-group}}(i))$

$n_{\text{key-group}}(i)$: the number of affected key-groups in CWGKM affected by i new CTEK(s).

$p_{\text{leader\_unit\_join}}$: the probability of user joining the member unit, $p_{\text{leader\_unit\_join}} = \frac{n_{\text{member\_in\_member\_units}}}{n_{\text{total\_group\_members}}}$

$p_1(\text{I})$: the probability of member unit leaving in Case I, $p_1(\text{I}) = \frac{n_{\text{member\_in\_member\_units}}}{n_{\text{total\_group\_members}}}$

$p_2(\text{I})$: the probability of leadership candidate leaving in Case I, $p_2(\text{I}) = \frac{n_{\text{leadership\_candidates}}}{n_{\text{total\_group\_members}}}$

$p_1(\text{II})$: the probability of member unit leaving in Case II, $p_1(\text{II}) = \frac{n_{\text{member\_in\_member\_units}}}{n_{\text{total\_group\_members}}}$

α : the degree of (α, L, x)-tree.

L : the level of (α, L, x)-tree.

x_i : the number of members in the node i.

$n_{\text{leadership\_candidates}}$: the total number of leadership candidates in CWGKM.

h_{unit} : the height of the key tree for the operation unit in CWGKM.

$n_{\text{TEK}}(\text{affected})$: the number of affected TEKs that need to be updated.

$n_{\text{key-groups}}$: the number of key-groups in CWGKM

j: the number of group applications affected by the join operation

l: the number of group applications affected by the leave operation

6.3.4 Key Independence

Key independence is a concept denoting that all keys in a group key management system are independent of each other and that no key can be derived from another key. In a key management system with keys possessing this characteristic, the disclosure of one key would not compromise the other keys. key independence is the foundation of other security properties of group key management such as backward and forward secrecy and prevention of collusion attacks.

The independence of keys is determined by a random key generation algorithm. In both CWGKM and TMKM, all the keys can be generated randomly and independently. Therefore, we consider that the keys in CWGKM and TMKM are independent.

6.3.5 Forward and Backward Secrecy

Forward secrecy prevents users who have left the group from accessing future group communication, while backward secrecy prevents new joining members from accessing former group content. Forward and backward secrecy are essential security properties provided by the group key management systems. Both can be achieved through key independence and by applying a rekeying procedure once membership change occurs. Based on probability theory [Allen, 1978; Grinstead & Laurie, 1991; Roberts, 1992], forward and backward secrecy can be ensured only if keys are independent.

To demonstrate forward secrecy, we assume that A refers to the old keys known to the departing group member *m* and B refers to the newly-generated keys that *m* wants to know. The probability A and B, $P(\mathrm{A})$ and $P(\mathrm{B})$, present the probability of A and B known to *m*. $P(\mathrm{A}) = 100\%$ and $P(\mathrm{B}) = 0\%$, because ex-member *m* knows A and does not know B. The conditional probability of deriving B by given A is denoted by:

$$P(\mathrm{B}|\mathrm{A}) = \frac{P(\mathrm{AB})}{P(\mathrm{A})}$$

If the key A and B are independent, then

$$P(\mathrm{AB}) = P(\mathrm{A})P(\mathrm{B})$$

Therefore,

$$P(\mathrm{B}|\mathrm{A}) = \frac{P(\mathrm{AB})}{P(\mathrm{A})} = \frac{P(\mathrm{A})P(\mathrm{B})}{P(\mathrm{A})} = P(\mathrm{B}) = 0\%$$

This result demonstrates that the probability of calculating the new keys from the old keys equals the probability of directly knowing the new keys. For the departing member, the probability of gaining the new keys through deriving from the old keys is zero because the ex-member does not know any new keys. Consequently, forward secrecy can be ensured because of the key independence.

In terms of backward secrecy, verification is similar to that of forward secrecy. When new member *m* joins the group application, we assume that A refers to the old keys which member *m* wants to know and B refers to the current keys which are known to *m*. The conditional probability of deriving A by given B is:

$$P(\mathrm{A}|\mathrm{B}) = \frac{P(\mathrm{AB})}{P(\mathrm{B})}$$

If the keys are independent, we can simplify the formula as:

$$P(\mathrm{A}|\mathrm{B}) = \frac{P(\mathrm{AB})}{P(\mathrm{B})} = \frac{P(\mathrm{A})P(\mathrm{B})}{P(\mathrm{B})} = P(\mathrm{A})$$

Because the old keys are unknown to the new joining member *m*, the conditional probability is:

$$P(\mathrm{A}|\mathrm{B}) = P(\mathrm{A}) = 0\%$$

The probability of obtaining the old keys from the current keys is zero. Backward secrecy is ensured due to the key independence.

In addition to key independence, in order to achieve forward and backward secrecy, keys also need to be updated immediately once membership change occurs. In CWGKM, a CKC enforces strict forward and backward secrecy by instantly invoking the rekeying process when a member joins or leaves the group application. Due to the application of multiple TEKs in CWGKM, key updating is able to be restricted within the cell. This reduces the impact of rekeying on the remaining group members.

On the other hand, in TMKM, the GKC also can perform key updating instantly to ensure forward and backward secrecy when membership change occurs. However, this secrecy is at the cost of operational efficiency because rekeying affects all remaining group members. As only one TEK is applied in TMKM, rekeying poses the 1-affect-n problem.

6.3.6 Prevention of Collusion Attacks

Collusion attack refers to any set of former group members being able to access the current TEK based on their knowledge of old keys. Collusion attack is a special case of forward secrecy, where a group of departing members works together with the intention of illegally regaining access to the current group key. As discussed in detail in section 6.3.4, the cause of collusion is rooted in key dependence, where new keys are generated by putting old keys through a one-way function. For example, in one-way function tree (OFT), the new key is generated by a one-way function. Two previous members can cooperate and share their old keys to generate a valid TEK within a period of time (described in section 2.2.2). Key dependency thus provides opportunities to the adversaries to compromise the new key if they know the old keys and the key generation function.

In both CWGKM and TMKM, all keys are generated randomly and independently. CWGKM and TMKM can therefore be considered immune to collusion attacks.

6.3.7 Trust Relationships

The security of keying materials is affected by the trust relationships among the parties involved in wireless group key management. In the wireless environment, there are at least two parties involved in key management: the wireless network operator and the service providers. In addition, several entities (base stations, GKCs and the local key controllers) from these two parties are involved in group key

management. In order to facilitate group key management, TMKM assumes that the network operator is trustable and invites the supervisor host (SH) and base stations, both from the network operator, to group key management. However, this trust relationship might not be acceptable in the real world. By inviting entities to group key management, TMKM provides the wireless network operator with the opportunity to access highly confidential keying materials, which poses the risk of compromising the secrecy of group communication. In contrast, CWGKM applies two level TEKs (group TEK and cell TEK) and double encryption to distinguish the security responsibilities of the various parties and entities. In CWGKM, group TEK is provided by the service provider for the sender(s) to encrypt the group communication at the application level, while cell TEK is applied by the wireless network operator to double encode the group content at a lower level before distributing it within the cell. This separation minimizes the need for a trust relationship between the service provider and the wireless network operator, because they do not need to know the TEKs of each other to perform key management.

6.3.8 Summary

In this section, we have analyzed and evaluated the proposed CWGKM. In comparison with TMKM, we have demonstrated that CWGKM offers performance and security advantages over TMKM on all assessment parameters. CWGKM can consequently be recognized as a practical, efficient and secure group key management solution to be deployed in the cellular wireless network. Table 6.4 summarizes the evaluation of CWGKM and TMKM.

Table 6.4 System evaluation of CWGKM and TMKM

	CWGKM	TMKM
Scalability	• Works with wireless network to cover large geographical areas • Supports various sized groups • Efficiently deals with highly dynamic groups	• Covers whole wireless network domain • Supports various sized groups • May be overwhelmed by highly dynamic membership changes
1-affect-n phenomenon	Minimizes the 1-affect-n problem by applying multiple TEKs	Suffers from the 1-affect-n phenomenon
Operational performance	• Applies micro-key management in operation units to achieve operational efficiency • Applies membership-oriented approach to efficiently address multiple-membership change issue	• Applies centralized key management which can cause operational inefficiency in communication and computation • Application-oriented structure cannot tackle multiple-membership change issue
Backward secrecy	Ensures strict backward secrecy by rekeying immediately when a user joins the group	Achieves strict backward secrecy
Forward secrecy	Achieves strict forward secrecy by invoking rekeying instantly once a member leaves the group	Achieves strict forward secrecy
Key independence	Each key is generated randomly and independently.	Keys are generated randomly and independently
Collusion prevention	Free of collusion attacks	Free of collusion attacks
Trust relationships	No trust relationships are required to be established between the involved parties	Full trust relationship needs to be established between involved parties.

6.4 Formulization of Group Key Management

In the previous chapters, we have analyzed and evaluated the operational costs of group key management based on a single member action such as join and leave. However, this kind of evaluation might not be sufficient to enable a system designer to estimate the capacity of key controllers, which need to be evaluated based on a large number of members and their behavior. In order to address this issue, in this section, we formulize the system performance of group key management systems from a statistical viewpoint. Section 6.4.1 introduces a formal model for the purpose of analysis and evaluation of the join and leave operations in a secure group application. From section 6.4.2 to section 6.4.4, we formulize the operations of join, group communication and leave respectively.

6.4.1 A Formal Model for Secure Group Applications

Although a single join and leave operation in a secure group application is considered a random action, some rules can nonetheless be identified based on the research involving a large number of users and their behavior in group applications. According to a study on the behavior of a large number of users on the multicast backbone network (MBone) [Almeroth & Ammar, 1996, 1997], in terms of general group applications that last from several hours to a day, the join and leave operations are the Poisson processes and the service time the members stays with the group communication is an exponential distribution. Based on these findings, the secure group application can be presented as a model (shown in Figure 6.10).

The whole process of a group application can be considered a "birth-death" process. For a user, the join operation is the start of a group communication and the leave operation is the end of a group communication. The life of a secure group application can therefore be divided into three stages (also shown in Figure 6.10).

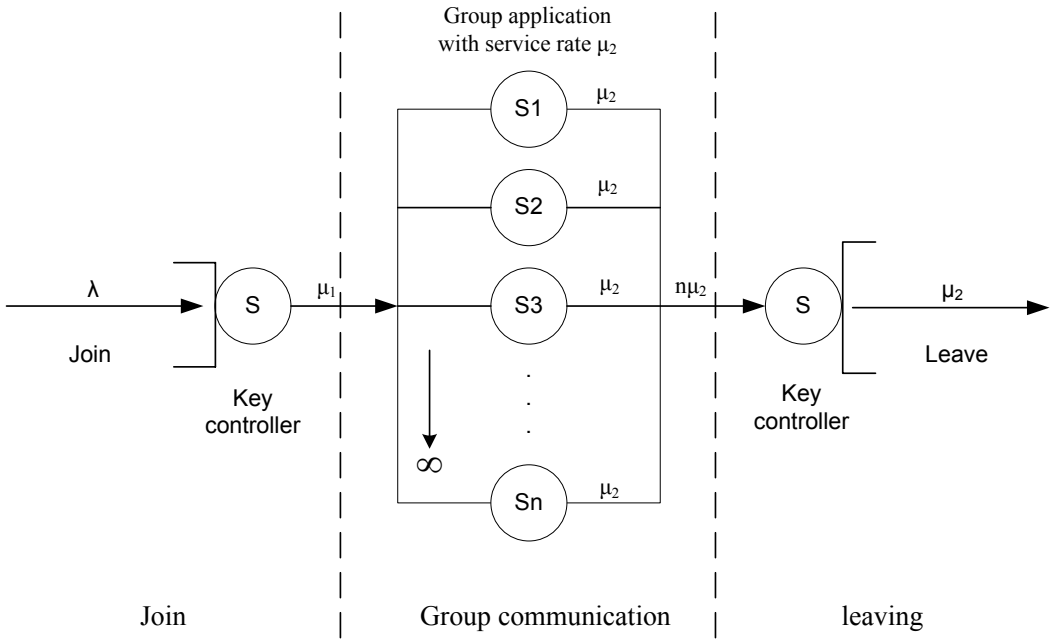


Figure 6.10 A formal model for secure group applications

- Join: a member joins a group by the Poisson process [Giambene, 2005; Robertazzi, 1994] and we assume that the processing time of join is exponential, because this assumption is realistic and common [Almeroth & Ammar, 1996; Sun & Trappe et al., 2004].
- Group communication: at this stage, members participate in the secure group application and the service time is an exponential distribution. During the communication, each member appears to be assigned to its own server, because

members can receive communication simultaneously by applying multicast transmission.

- Leave: a member leaves the group application at this stage. Based on Burke's theorem [Robertazzi, 1994], the departure process is also a Poisson process if the arrival is by the Poisson process and the service time is an exponential distribution. We assume that the leave operation also has an exponential service time.

In the following sections, we discuss each of these three stages in depth.

6.4.2 The Join Operation

In relation to a single user, each join operation is a random variable, but from a statistical viewpoint, member arrival is a Poisson process. In addition, we assume that the service time of a key controller are independent and exponentially distributed. Thus, we can use the M/M/1 queuing model [Leonard Kleinrock, 1975; Leonard Kleinrock & Gail, 1996; Robertazzi, 1994] to analyze the join operation. We define that the arrival rate of a user is λ and the service rate of the key controller is μ_1, as shown in Figure 6.10. Therefore, the possibility of n users in the system is:

$$P_n = \rho^n (1-\rho)$$

where $\rho = \lambda / \mu_1$, known as utilization.

We can then calculate the average number of members arriving in the key controller during a period of time by applying the theory of expectation value.

$$\overline{n} = E(n) = \sum_{n=0}^{\infty} nP_n = \sum_{n=0}^{\infty} n(1-\rho)\rho^n = (1-\rho)\sum_{n=0}^{\infty} n\rho^n$$

And

$$\sum_{n=0}^{\infty} n\rho^n = \rho \sum_{n=1}^{\infty} n\rho^{n-1} = \rho \frac{d}{d\rho} \sum_{n=1}^{\infty} \rho^n = \rho \frac{d}{d\rho} \frac{\rho}{1-\rho} = \frac{\rho}{(1-\rho)^2}$$

Thus, we can obtain that

$$\overline{n} = \frac{\rho}{1-\rho}$$

We define that the operational cost of the join operation in terms of the whole group and a single member are $E(\text{join})$ and $C_{\text{join}}(\text{single})$. The operational cost of the join operation for a group key management system is thus:

$$E(\text{join}) = \sum_{n=0}^{\infty} nP_n \times C_{\text{join}}(\text{single}) = \frac{\rho \times C_{\text{join}}(\text{single})}{1-\rho}$$

where $C_{\text{join}}(\text{single})$ can be the operational cost of communication, computation and key storage, as discussed in Chapter 4 and 5 and summarized in Table 6.3.

6.4.3 Group Communication

In the stage of group communication, each arriving member appears to be assigned to a dedicated server. Thus, we can apply the M/M/∞ model [Giambene, 2005; Robertazzi, 1994] in queuing theory to analyze this stage. We assume that the member arrival rate is μ_1; that is, the service rate of the key controller in the join stage. We also assume the service rate of multicast is μ_2, as illustrated in Figure 6.10. Based on the M/M/∞ model, the possibility of *n* members in the system is:

$$P_n = \frac{1}{n!}\left[\frac{\mu_1}{\mu_2}\right]^n e^{-\mu_1/\mu_2}$$

Therefore, the average number of members in the group application is:

$$\overline{n} = \sum_{n=0}^{\infty} nP_n = \frac{\mu_1}{\mu_2}$$

6.4.4 The Leave Operation

Similar to the join operation, a single leave operation is also a random variable. However, based on Burke's theorem [Robertazzi, 1994] and from statistical viewpoint, the leave operation is also a Poisson process if the arrival is the Poisson process and the service time is the exponential distribution. We assume that the processing time rate for leave is μ_3. In addition, the arrival rate for the leave is $n\mu_2$ (shown in Figure 6.10). Therefore, we can apply the M/M/1 queuing model to calculate that the possibility of *n* members in the key controller for departure is:

$$P_n = \rho^n (1-\rho)$$

where we define $\rho = \frac{n\mu_2}{\mu_3}$.

Based on the above formula, we can calculate the average number of members arriving in the key controller for leave during a period of time by applying the theory of expectation value.

$$\overline{n} = E(n) = \sum_{n=0}^{\infty} nP_n = \sum_{n=0}^{\infty} n(1-\rho)\rho^n = (1-\rho)\sum_{n=0}^{\infty} n\rho^n$$

In a similar way to the join process, we can obtain that:

$$\overline{n} = \frac{\rho}{1-\rho}$$

We define that the operational cost of the leave operation in terms of the whole group and a single member are $E(\text{leave})$ and $C_{\text{leave}}(\text{single})$. The operational cost of the leave operation for the group key management system is thus:

$$E(\text{leave}) = \sum_{n=0}^{\infty} nP_n \times C_{\text{leave}}(\text{single}) = \frac{\rho \times C_{\text{leave}}(\text{single})}{1-\rho}$$

where $C_{\text{leave}}(\text{single})$ is the operational cost of communication, computation and key storage for the leave operation, as discussed in detail in Chapters 4 and 5 and summarized in Table 6.3.

6.4.5 Summary

In this section, we have formulized the join and leave operations in secure group applications from a statistical viewpoint. This formulization gives system designers the means to analyze, evaluate and estimate the performance and capacity of group key management systems. Table 6.5 tabulates the formulas of the join and leave operations.

Table 6.5 The operational cost of the join and leave operations

	join	leave
The average number of members in the system ($\overline{n}$)	$\overline{n} = \frac{\rho}{1-\rho}$ $(\rho = \lambda / \mu_1)$	$\overline{n} = \frac{\rho}{1-\rho}$ $(\rho = \frac{n\mu_2}{\mu_3})$
The cost of operation (E)	$E(\text{join}) = \frac{\rho \times C_{\text{join}}(\text{single})}{1-\rho}$	$E(\text{leaving}) = \frac{\rho \times C_{\text{leaving}}(\text{single})}{1-\rho}$

λ : the arriving rate of members to join the group application
μ_1 : the processing rate of the key controller for join
μ_2 : the service rate of multicast
μ_3 : the processing rate of key controller for leaving
$C_{\text{join}}(\text{single})$: the operational costs of a member's join in terms of communication, computation and key storage
$C_{\text{leave}}(\text{single})$: the operational cost of a member's leave in terms of communication, computation and key storage

6.5 Summary

In this chapter, we have integrated the proposed three group key management approaches discussed in depth in Chapters 3, 4 and 5 into a CWGKM solution for the cellular wireless network. We have used a case study to demonstrate the operations of this solution in detail. In addition, we have analyzed and evaluated the integrated solution based on the assessment parameters described in section 3.3.2. In comparison with TMKM, CWGKM provides a number of advantages in the areas of performance and security. CWGKM exhibits better performance than TMKM especially in terms of multiple-membership key management because CWGKM uses a membership-oriented approach to re-organize key management structure to tackle multiple-membership changes. Overall, CWGKM can be considered an efficient, practical and secure design suitable for deployment in the cellular wireless network.

Chapter 7

Conclusion

Group key management is essential to ensure the security of group applications in a wireless network environment. The open structure of IP multicast transmission and wireless networks renders access control problematic. In order to secure group communication, a shared group key must be applied to encrypt the communication data. The security of the group communication therefore depends on the safety of the group key. Group key management consequently plays a critical role in the security of group communication.

The resource limitations of both wireless networks and the mobile devices have ramifications for group key management in the areas of performance, security and network compatibility. In order to tackle these problems, we focused on three levels of research in this thesis: the formal model level, the system component level and the system solution level.

At the first level, a formal wireless group key management system model has been proposed. This model not only identifies the important and essential

components in the wireless group key management system, but also clarifies the relationships between the problems in wireless group key management and these components. These key components and relationships provide the building blocks and methods to design a system for the purpose of tackling the problems of wireless group key management. In order to analyze and evaluate wireless group key management approaches, we have been developed a set of assessment parameters pertaining to the three problems (performance, security and network compatibility) of wireless group key management. The proposed model and parameters can serve as a guideline for system designers and implementers to design, analyze and evaluate wireless group key management systems.

At the system component level, based on the proposed model, we have proposed three group key management approaches: (i) a group key management architecture for the cellular wireless network; (ii) hybrid group key management (HGKM); and (iii) membership-oriented key management (MOKM). Each of these is designed to tackle a particular group key management problem in the wireless network, as follows.

- A group key management architecture for the cellular wireless network

In the wireless environment, group members are spread across a large geographic area and the membership is highly dynamic. Any membership change would cause the 1-affect-n phenomenon. In other words, key updating caused by a membership change would affect all remaining members in the group. In order to tackle this scalability and the 1-affect-n problem, we have proposed a group key management architecture for the cellular wireless network. The major features of this

architecture are as follows.

- It is tailored to the cellular wireless network to address network compatibility issues.
- It seamlessly integrates key management architecture with the cellular wireless network infrastructure to provide geographic scalability.
- It copes with various group sizes and frequent membership changes by applying distributed key management.
- It applies multiple TEKs to restrict the impact of rekeying on the group members to minimize the 1-affect-n phenomenon.

• Hybrid group key management (HGKM)

Due to the resource limitations of both wireless networks and mobile devices, operational performance is the highest priority of a wireless group key management system. In order to reduce the operational costs of communication, computation and key storage for key management in the cellular wireless network, a hybrid group key management (HGKM) approach has been proposed. Micro-key management is introduced in HGKM to achieve operational efficiency. In HGKM, the group key management structure is divided into a number of small sections called operation units. Key management operations are performed based on the operation units. By performing micro-key management, rekeying can be confined within the scope of an operation unit. This reduces the operational costs of key management. Compared to the current group key management schemes, LKH and OFT, HGKM can achieve greater performance in the areas of communication, computation and key storage.

- Membership-oriented key management (MOKM)

Multiple-membership is a common scenario in secure group applications. In the traditional application-oriented group key management approaches, members having multiple memberships need to register with several separate key management structures where each key management structure is associated with a group application. This approach does not offer operational efficiency, because redundant members' information exists in the key management structures. In addition, several key management structures are affected when multiple-membership change occurs. In order to address this issue, we have proposed a membership-oriented key management (MOKM) approach. Compared to the application-oriented approach, MOKM offers more efficient key management where multiple-membership is involved. The efficiencies are made possible through two MOKM features: a multiple-membership key-group, and a key management structure based on user memberships.

- Multiple-membership key-group

  A new key management structure, called multiple-membership key-group, is introduced in MOKM to accommodate members with multiple memberships. A member with multiple memberships is assigned into a proper multiple-membership key-group according to its memberships when the member joins the group applications. This new structure eliminates redundant registration in the key management structures.

- Key management structure based on user memberships

  In MOKM, the key management structure is organized based on the user memberships. A member is assigned into one and only one key-group according to its membership(s), no matter how many memberships it has. Therefore, only one key management structure is affected during the rekeying process. Compared with application-oriented key management approaches, MOKM reduces the operational costs of rekeying when multiple-membership change occurs.

At the system solution level, we have proposed a comprehensive group key management solution for the cellular wireless network (CWGKM). This is achieved by integrating the three group key management approaches (group key management architecture for the cellular wireless network, HGKM and MOKM) described at the system component level. Based on the assessment parameters proposed at the first level (the formal model level) of this research, this integrated solution satisfies all the performance and security requirements of a wireless group key management system.

In order to design a wireless group key management system, system designers need to analyze and evaluate the capacity of the system in relation to the behavior of a large number of members. However, analysis based on a single user's operation cannot fulfill this task. Therefore, at the system solution level, we have formulized the users' actions of join and leave according to the queuing theory. This formulization provides a powerful approach for system designers to perform quantitative analysis and evaluation of a group key management system.

In terms of future research, we plan to apply group key management to the wireless sensor network [Callaway, 2004; Perrig & John et al., 2004; Walters & Liang et al., 2006]. The wireless sensor network is gaining acceptance and popularity due to its potentially low-cost solutions to a variety of real-world challenges both in military and civilian areas. Future research in this area could adjust current group key management approaches to this new environment or develop new group key management schemes dedicated to sensor networks. Future research could also apply group key management to banking and financial systems to secure the internet banking system, thereby protecting users and helping to minimize online fraud [Dandash & Wang et al., 2007a, 2007b, 2008].

In conclusion, this study suggests key management solutions to efficiency and security problems associated with group applications in wireless networks. The results justify continuing research in this area in order to increase the body of knowledge surrounding group key management systems, and to provide practical support to those involved in developing key management systems for secure group applications in wireless networks.

References

[Ahson & Ilyas, 2007] Ahson, S. A., and Ilyas, M. (2007). *WiMAX Applications*. CRC press.

[Akyildiz & McNair et al., 1998] Akyildiz, I. F., McNair, J., Ho, J., Uzunalioglu, H., and Wang, W. (1998). *Mobility Management in Current and Future Communications Networks*. IEEE Network, Vol. 12, pp. 39-49.

[Albanna & Almeroth et al., 2001] Albanna, Z., Almeroth, K., Meyer, D., and Schipper, M. (2001). *IANA Guidelines for IPv4 Multicast Address Assignments*. RFC 3171.

[Allen, 1978] Allen, A. O. (1978). *Probability, Statistics, and Queueing Theory with Computer Science Applications*. Academic Press.

[Almeroth & Ammar, 1996] Almeroth, K. C., and Ammar, M. H. (1996). *Collecting and Modeling the Join/Leave Behavior of Multicast Group Members in the MBone.* In Proceedings of the International Symposium on High Performance Distributed Computing (HPDC), pp. 209-216.

[Almeroth & Ammar, 1997] Almeroth, K. C., and Ammar, M. H. (1997). *Multicast Group Behavior in the Internet's Multicast Backbone (MBone)*. IEEE Communications Magazine, Vol. 35(6), pp. 224-229.

[Amir & Kim et al., 2004] Amir, Y., Kim, Y., Nita-Rotaru, C., Schultz, J. L., Stanton, J., and Tsudik, G. (2004). *Secure Group Communication Using Robust Contributory Key Agreement*. IEEE Transactions on Parallel and Distributed Systems, Vol. 15(5), pp. 468-480.

[Ballardie, 1996] Ballardie, T. (1996). *Scalable Multicast Key Distribution*. RFC 1949.

[Banerjee & Bhattacharjee et al., 2002] Banerjee, S., Bhattacharjee, B., and Kommareddy, C. (2002). *Scalable Application Layer Multicast* In Proceedings of the ACM SIGCOMM'02, pp. 205-217.

[Baugher & Canetti et al., 2005] Baugher, M., Canetti, R., Dondeti, L. R., and Fredrik, L. (2005). *Multicast Security Group Key Management Architecture*. RFC 4046.

[Becker & Wille, 1998] Becker, K., and Wille, U. (1998). *Communication Complexity of Group Key Distribution.* In Proceedings of the the 5th ACM Conference on Computer and Communications Security pp. 1-6.

[Briscoe, 1999] Briscoe, B. (1999). *MARKS: Zero Side Effect Multicast Key Management using Arbitrarily Revealed Key Sequences.* In Proceedings of the First International Workshop on Networked Group Communication, pp. 301-320.

[Brown & Singh, 1998] Brown, K., and Singh, S. (1998). *RelM:Reliable Multicast for Mobile Networks*. Computer Communication, Vol. 21(16), pp. 1379-1400.

[Bruschi & Rosti, 2002] Bruschi, D., and Rosti, E. (2002). *Secure Multicast in Wireless Networks of Mobile Hosts: Protocols and Issues*. Mobile Networks and Applications, Vol. 7(6), pp. 503-511.

[Cain & Deering et al., 2002] Cain, B., Deering, S., Kouvelas, I., Fenner, B., and Thyagarajan, A. (2002). *Internet Group Management Protocol, Version 3*. RFC 3376.

[Callaway, 2004] Callaway, E. H. (2004). *Wireless Sensor Networks : Architectures and Protocols*. Auerbach Publications.

[Canetti & Garay et al., 1999] Canetti, R., Garay, J., Itkis, G., Micciancio, D., Naor, M., and Pinkas, B. (1999). *Multicast Security: A taxonomy and Some Efficient Constructions.* In Proceedings of the Eighteenth Annual Joint Conference of the IEEE Computer and Communications Societies., pp. 708-716.

[Caronni & Waldvogel et al., 1998] Caronni, G., Waldvogel, K., Sun, D., and Plattner, B. (1998). *Efficient Security for Large and Dynamic Multicast Groups.* In Proceedings of the Enabling Technologies: Infrastructure for Collaborative Enterprises, 1998. (WET ICE '98) Proceedings., Seventh IEEE International Workshops on, pp. 376-383.

[Challal & Bettahar et al., 2004] Challal, Y., Bettahar, H., and Bouabdallah, A. (2004). *SAKM: A Scalable and Adaptive Key Management Approach for Multicast Communications*. ACM SIGCOMM Computer Communications Review, ,Vol. 34(2)(2), pp. 55-70.

[Challal & Seba, 2005] Challal, Y., and Seba, H. (2005). *Group Key Management Protocols:A Novel Taxonomy*. International Journal of Information Technology,Vol. 2(1), pp. 105-118.

[Chen & Zhang, 2004] Chen, J.-C., and Zhang, T. (2004). *IP-Based Next-Generation Wireless Networks*. John wiley & Sons, Inc.

[Chen & Wang et al., 2006] Chen, Y., Wang, Y., Wu, X., and Le, P. D. (2006). *The Design of Cluster-based Group Key Management System in Wireless Networks.* In Proceedings of the International Conference on Communication Technology (ICCT 2006), pp. 209-212.

[Chichester & Hoboken, 2008]Chichester, E., and Hoboken, N. J. (2008). *Mobile WiMAX*. John Wiley & Son, Inc.

[Dandash & Wang et al., 2007a] Dandash, O., Wang, Y., Le, P. D., and Srinivasan, B. (2007a). *A New Dynamic Key Generation Scheme for Fraudulent Internet Payment Prevention.* In Proceedings of the Fourth International Conference on Information Technology (ITNG '07), pp. 83-88.

[Dandash & Wang et al., 2007b] Dandash, O., Wang, Y., Le, P. D., and Srinivasan, B. (2007b). *A New Group Key Management Structure for Fraudulent Internet Banking Payments Detection.* In Proceedings of the the 9th International Conference on Enterprise Information Systems (ICEIS 2007), pp. 57-62.

[Dandash & Wang et al., 2008] Dandash, O., Wang, Y., Le, P. D., and Srinivasan, B. (2008). *Fraudulent Internet Banking Payments Prevention using Dynamic Key*. Journal of Networks,Vol. 3(1), pp. 25-34.

[DeCleene & Dondeti et al., 2001] DeCleene, B., Dondeti, L., Griffin, S., Hardjono, T., Kiwior, D., Kurose, J., et al. (2001). *Secure Group Communications for Wireless Networks.* In Proceedings of the Military Communications Conference, 2001. MILCOM 2001. Communications for Network-Centric Operations: Creating the Information Force. IEEE, pp. 113-117.

[Deering, 1988] Deering, S. E. (1988). *Multicast Routing in Internetworks and Extended LANs*. ACM Special Interest Group on Data Communication (ACM SIGCOMM), pp. 55-64.

[Deering, 1989] Deering, S. E. (1989). *Host Extensions for IP Multicasting*. RFC 1112.

[Diffie & Hellman, 1976] Diffie, W., and Hellman, M. E. (1976). *Multiuser cryptographic techniques.* In Proceedings of the AFIPS, pp. 109-112.

[Dondeti & Mukherjee et al., 2000] Dondeti, L. R., Mukherjee, S., and Samal, A. (2000). *Scalable Secure One-to-Many Group Communication Using Dual Encryption*. Computer Communications, Vol. 23(17), pp. 1681-1701.

[Du & Ni et al., 1999] Du, F., Ni, L. M., and Esfahanian, A.-H. (1999). *Towards Solving Multicast Key Management Problem.* In Proceedings of the Eighth International Conference on Computer Communications and Networks (ICCCN'99), pp. 232-236.

[Fenner, 1997] Fenner, W. C. (1997). *Internet Group Management Protocol version 2*. RFC 2236.

[Ghosh & Anjum, 2005] Ghosh, A., and Anjum, F. (2005). *Last Hop Topology Sensitive Multicasting Key Managment.* In Proceedings of the the 1st ACM international workshop on Quality of service & security in wireless and mobile networks, pp. 63-70.

[Giambene, 2005] Giambene, G. (2005). *Queuing Theory and Telecommunications Networks and Applications*. Springer Science+Business Media, Inc.

[Glisic, 2006] Glisic, S. G. (2006). *Advanced Wireless Networks : 4G Technologies*. John Wiley Inc.

[Goncalves & Niles, 1999] Goncalves, M., and Niles, K. (1999). *IP multicasting Concepts and Applications*. McGraw-Hill Companies, Inc.

[Gong & Shacham, 1995] Gong, L., and Shacham, N. (1995). *Multicast Security and its Extensions to a Mobile Environment*. ACM-Baltzer Journal of Wireless Networks, Vol. 1(3), pp. 281-295.

[Gossain & Cordeiro et al., 2002] Gossain, H., Cordeiro, C. d. M., and Agrawal, D. (2002). *Multicast: Wired to Wireless*. IEEE Communications Magazine, Vol. 40(6), pp. 116-123.

[Grinstead & Laurie, 1991] Grinstead, C. M., and Laurie, S. J. (1991). *Introduction to Probability (2nd Edition)*. American Mathematical Society.

[Hardjono & Baugher et al., 2001] Hardjono, T., Baugher, M., and Harney, H. (2001). *Group Key Management for IP Multicast: Model & Architecture.* In Proceedings of the Tenth IEEE Enabling Technologies: Infrastructure for Collaborative Enterprises, pp. 223-228.

[Hardjono & Cain et al., 2000] Hardjono, T., Cain, B., and Monga, I. (2000). *Intra-Domain Grouop Key Management Protocol*. IETF draft.

[Hardjono & Dondeti, 2003] Hardjono, T., and Dondeti, L. R. (2003). *Multicast and Group Security*. Artech House, Inc.

[Hardjono & Harney, 2002] Hardjono, T., and Harney, H. (2002). *Group Security Policy Management for IP Multicast and Group Security.* In Proceedings of the the Second International IFIP-TC6 Networking Conference on Networking Technologies, Services, and Protocols, pp. 1123-1128.

[Hardjono & Tsudik, 1999] Hardjono, T., and Tsudik, G. (1999). *IP Multicast Security: Issues and Directions*: Annales de Telecom.

[Harkins & Carrel, 1998] Harkins, D., and Carrel, D. (1998). *The Internet Key Exchange (IKE)*. RFC 2409.

[Harney & Colgrove et al., 2001]Harney, H., Colgrove, A., and McDaniel, P. (2001). *Principles of Policy in Secure Groups.* In Proceedings of the Network and Distributed Systems Security Internet Society, pp. 125-135.

[Harney & Harder, 1999] Harney, H., and Harder, E. (1999). *Logical Key Hierarchy Protocol*. Internet-Draft, Internet Engineering Task Force (IETF).

[Harney & Muckenhirn, 1997a] Harney, H., and Muckenhirn, C. (1997a). *Group Key Management Protocol (GKMP) Architecture*. RFC 2094.

[Harney & Muckenhirn, 1997b] Harney, H., and Muckenhirn, C. (1997b). *Group Key Management Protocol (GKMP) Specification*. RFC 2093.

[Holbrook & Cheriton, 1999] Holbrook, H. W., and Cheriton, D. R. (1999). *IP Multicast Channels: EXPRESS Support for Large-Scale Single-Source Applications.* In Proceedings of the ACM Special Interest Group on Data Communication (SIGCOMM' 99), pp. 65 - 78.

[Horng, 2002]Horng, G. (2002). *Cryptanalysis of A Key Management Scheme for Secure Multicast Communications*. IEICE Transactions on Communications Vol. E85-B(5), pp. 1050-1051.

[Hosseini & Ahmed et al., 2007] Hosseini, M., Ahmed, D. T., Shirmohammadi, S., and Georganas, N. D. (2007). *A Survey of Application-Layer Multicast Protocols*. IEEE Communications Surverys & Tutorials, Vol. 9(3), pp. 58-74.

[IANA, 2008]IANA. (2008). Internet Multicast Addresses. from Internet Assigned Numbers Authority, http://www.iana.org/assignments/multicast-addresses.

[Judge & Ammar, 2003] Judge, P., and Ammar, M. (2003). *Security Issues and Solutions in Multicast Content Distribution: A Survey*. Network, IEEE, Vol. 17(1), pp. 30-36.

[Kaufman, 2005] Kaufman, C. (2005). *Internet Key Exchange (IKEv2) Protocol*. RFC 4306.

[Kim & Perrig et al., 2001] Kim, Y., Perrig, A., and Tsudik, G. (2001). *Communication-Efficient Group Key Agreement.* In Proceedings of the 17th Internation Information Security Conference, pp. 229-244.

[Kim & Perrig et al., 2004a] Kim, Y., Perrig, A., and Tsudik, G. (2004a). *Group Key Agreement Efficient in Communication*. IEEE Transactions on Computers, Vol. 53(7), pp. 905-921.

[Kim & Perrig et al., 2004b] Kim, Y., Perrig, A., and Tsudik, G. (2004b). *Tree-Based Group Key Agreement*. ACM Transactions on Information and System Security, Vol. 7(1), pp. 60-96.

[Kleinrock, 1975] Kleinrock, L. (1975). *Queueing Systems* (Vol. 1). Wiley-Interscience.

[Kleinrock & Gail, 1996] Kleinrock, L., and Gail, R. (Eds.). (1996). *Queueing Systems: Problems and Solutions*. New York: J. Wiley.

[Kruus, 1998] Kruus, P. S. (1998). *A survey of Multicast Security Issues and Architectures.* In Proceedings of the In the 21st National Information Systems Security Conference, pp. 408-420.

[Ku & Chen, 2003] Ku, W.-C., and Chen, S.-M. (2003). *An Improved Key Management Scheme for Large Dynamic Groups Using One-Way Functions Trees.* In Proceedings of the International Conference on Parallel Processing Workshops, pp. 391-396.

[Li & Yang et al., 2001] Li, X. S., Yang, Y. R., Gouda, M. G., and Lam, S. S. (2001). *Batch Rekeying for Secure Group Communications.* In Proceedings of the 10th International World Wide Web Conference (WWW10), pp. 525-534.

[MacDonald, 1979] MacDonald, V. H. (1979). *The Cellular Concept*. The Bell System Technical Journal, Vol. 58(1), pp. 15-41.

[Maughan & Schneider et al., 1998] Maughan, D., Schneider, M., Schertler, M., and Turner, J. (1998). *Internet Security Association and Key Management Protocol (ISAKMP)*. RFC 2408.

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

[McHugh & Michale, 1999] McHugh, J., and Michale, B. J. (1999). *Secure Group Management in Large Distributed Systems: What Is A Group and What Does It Do?* In Proceedings of the Workshop on New security paradigms, pp. 80-85.

[Mittra, 1997] Mittra, S. (1997). *Iolus: A Framework for Scalable Secure Multicasting.* In Proceedings of the ACM SIGCOMM, pp. 277-288.

[Moyer & Rao et al., 1999] Moyer, M. J., Rao, J. R., and Rohatgi, P. (1999). *A Survey of Security Issues in Multicast Communications*. Network, IEEE, Vol. 13(6), pp. 12-23.

[O'Driscoll, 2008] O'Driscoll, G. (2008). *Next Generation IPTV services and Technologies*. John Wiley & Sons, Inc.

[Pahlavan & Krishnamurthy, 2001] Pahlavan, K., and Krishnamurthy, P. (2001). *Principles of Wireless Networks: A Unified Approach*. Prentice Hall PTR.

[Pegueroles & Rico-Novella et al., 2003] Pegueroles, J., Rico-Novella, F., Hernandez-Serrano, J., and Soriano, M. (2003). *Improved LKH for Batch Rekeying in Multicast Groups.* In Proceedings of the International Conference on Information Technology: Research and Education (ITRE2003), pp. 269-273.

[Perrig & John et al., 2004] Perrig, A., John, S., and Wagner, D. (2004). *Security in Wireless Sensor Networks*. Communications of the ACM, Vol. 47(6), pp. 53-57.

[Perrig & Song et al., 2001] Perrig, A., Song, D., and Tygar, J. D. (2001). *ELK, a New Protocol for Efficient Large-Group Key Distribution.* In Proceedings of the IEEE Symposium on Security and Privacy, pp. 247-262.

[Poole, 2006] Poole, I. (2006). *Cellular Communications Explained from Basics to 3G*. Elsevier Ltd.

[Radha Krishna Rao & Radhamani, 2008] Radha Krishna Rao, G. S. V., and Radhamani, G. (2008). *WiMAX : A Wireless Technology Revolution*. Auerbach Publications.

[Rafaeli & Hutchison, 2002]Rafaeli, S., and Hutchison, D. (2002). *Hydra: A Decentralized Group Key Management.* In Proceedings of the 11th IEEE International Workshops on Enabling Technologies: Infrastructure for Collaborative Enterprises, pp. 62-67.

[Rafaeli & Hutchison, 2003]Rafaeli, S., and Hutchison, D. (2003). *A Survey of Key Management for Secure Group Communications*. ACM Computing Surveys (CSUR), Vol. 35(3), pp. 309-329.

[Rhee, 1998] Rhee, M. Y. (1998). *CDMA Cellular Mobile Communications & Network Security*. Prentice Hall PTR.

[Robertazzi, 1994] Robertazzi, T. G. (1994). *Computer Networks and Systems: Queuing Theory and Performance Evaluation* (2nd ed.). Springer-Verlag New York, Inc.

[Roberts, 1992] Roberts, R. A. (1992). *An Introduction to Applied Probability*. Addison-Wesley Publishing Co. Inc.

[Rodeh & Birman et al., 2000] Rodeh, O., Birman, K. P., and Dolev, D. (2000). *Optimized Group Rekey for Group Communication Systems.* In Proceedings of the Network and Distributed System Security, pp. 39-48.

[Salkintzis, 2004] Salkintzis, A. K. (2004). *Mobile Internet Enabling Technologies and Services*. CRC press LLC.

[Setia & Koussih et al., 2000] Setia, S., Koussih, S., and Jajodia, S. (2000). *Kronos: A Scalable Group Re-Keying Approach for Secure Multicast.* In Proceedings of the IEEE Symposium on Security and Privacy, pp. 215-228.

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

[Steiner & Tsudik et al., 1996] Steiner, M., Tsudik, G., and Waidner, M. (1996). *Diffie-Hellman Key Distribution Extended to Group Communication.* In Proceedings of the 3rd ACM Conference on Computer and Communications Security, pp. 31-37.

[Steiner & Tsudik et al., 2000] Steiner, M., Tsudik, G., and Waidner, M. (2000). *Key Agreement in Dynamic Peer Groups*. IEEE Transactions on Parallel and Distributed Systems, Vol. 11(8), pp. 769-780.

[Sun & Trappe et al., 2002] Sun, Y., Trappe, W., and Liu, K. J. R. (2002). *An Efficient Key Management Scheme for Secure Wireless Multicast.* In Proceedings of the IEEE International Conference on Communication (ICC'02), pp. 1236-1240.

[Sun & Trappe et al., 2003] Sun, Y., Trappe, W., and Liu, K. J. R. (2003). *Topology-Aware Key Management Schemes for Wireless Multicast.* In Proceedings of the Global Telecommunications Conference, pp. 1471-1475.

[Sun & Trappe et al., 2004] Sun, Y., Trappe, W., and Liu, K. J. R. (2004). *A Scalable Multicast Key Management Scheme for Heterogeneous Wireless Networks*. IEEE/ACM Transactions on Networking, Vol. 12(4), pp. 653-666.

[Thajchayapong & Peha, 2006] Thajchayapong, S., and Peha, J. M. (2006). *Mobility Patterns in Microcellular Wireless Networks*. IEEE Transactions on Mobile Computing, Vol. 5(1), pp. 52-63.

[Varshney, 2002] Varshney, U. (2002). *Multicast Support in Mobile Commerce Applications*. Computer, Vol. 35(2), pp. 115-117.

[Viterbi, 1995] Viterbi, A. J. (1995). *CDMA: Principles of Spread Spectrum Communication*. Addison-Wesley Publishing Co. Inc.

[Waldvogel & Caronni et al., 1999] Waldvogel, M., Caronni, G., Sun, D., Weiler, N., and Plattner, B. (1999). *The VersaKey Framework: Versatile Group Key Management*. IEEE Journal on Selected Areas in Communications Vol. 17(9), pp. 1614-1631.

[Wallner & Harder et al., 1999] Wallner, D., Harder, E., and Agee, R. (1999). *Key Management for Multicast: Issues and Architectures*. RFC 2627.

[Walters & Liang et al., 2006] Walters, J. P., Liang, Z., Shi, W., and Chaudhary, V. (2006). *Security in Distributed, Grid and Pervasive Computing*. Auerbach Publicatons, CRC press.

[Wang & Damodaran et al., 2006] Wang, Y., Damodaran, D., and Le, P. D. (2006). *Efficient Group Key Management in Wireless Networks.* In Proceedings of the 3rd International Conference on Information Technology: New Generations (ITNG 2006), pp. 432-437.

[Wang & Le, 2005a] Wang, Y., and Le, P. D. (2005a). *Secure Group Communications in Wireless Networks.* In Proceedings of the 3rd International Conference on Advances in Mobile Multimedia (MoMM2005), pp. 241-252.

[Wang & Le, 2005b] Wang, Y., and Le, P. D. (2005b). *Scalable Multi-Subgroup Key Management in Wireless Networks*. International Journal of Computer Science and Network Security, Vol. 5(11), pp. 95-105.

[Wang & Le, 2007a] Wang, Y., and Le, P. D. (2007a). *Efficient and Scalable Group Key Management in Wireless Networks* Encyclopedia of Mobile Computing & Commerce Vol. 2, pp. 832-838. IGI Publishing.

[Wang & Le, 2007b] Wang, Y., and Le, P. D. (2007b). *Secure Group Communications in Wireless Networks* Encyclopedia of Mobile Computing & Commerce Vol. 1, pp. 227-232. IGI Publishing.

[Wang & Le et al., 2007] Wang, Y., Le, P. D., and Srinivasan, B. (2007). *Hybrid Group Key Management Scheme for Secure Wireless Multicast.* In Proceedings of the 6th IEEE International Conference on Computer and Information Science (ICIS 2007), pp. 346-351.

[Weiler, 2001] Weiler, N. (2001). *Semsomm - A Scalable Multiple Encryption Scheme for One-to-Many Multicast.* In Proceedings of the 10th IEEE International Workshops on Enabling Technologies: Infrastructure for Collaborative Enterprises (WET ICE 2001), pp. 231-236.

[Wong & Gouda et al., 1998] Wong, C. K., Gouda, M., and Lam, S. S. (1998). *Secure group communications using key graphs.* In Proceedings of the ACM SIGCOMM, pp. 68-79.

[Wong & Gouda et al., 2000] Wong, C. K., Gouda, M., and Lam, S. S. (2000). *Secure Group Communications Using Key Graphs*. IEEE/ACM Transactions on Networking, Vol. 8(1), pp. 16-30.

[Yang & Li et al., 2001] Yang, Y. R., Li, S. X., Zhang, B. X., and Lam, S. S. (2001). *Reliable Group Rekeying: A Performance Analysis.* In Proceedings of the 2001 Conference on Applications, Technologies, Architectures, and Protocols for Computer Communications, pp. 27-38.

[Zappala & Lo et al., 2004] Zappala, D., Lo, V., and GauthierDickey, C. (2004). *The Multicast Address Allocation Problem: Theory and Practice*. Computer Networks, Vol. 45(1), pp. 55-73.

[Zhang & Jamin et al., 2002] Zhang, B., Jamin, S., and Zhang, L. (2002). *Host Multicast: A Framework for Delivering Multicast to End Users.* In Proceedings of the IEEE Conference on Computer Communications (INFOCOM'02), pp. 1366-1375.

[Zou & Ramamurthy et al., 2005] Zou, X., Ramamurthy, B., and Magliveras, S. S. (2005). *Secure Group Communications over Data Networks*. Springer Science+Business Media, Inc.

ERRATA

p 173 para 2, line 5: "$\sum_{i=1}^{n} C_n^i = 2^n - 1$" for "$\sum_{i=2}^{n} C_n^i$"

p 97 Figure 4.1: "dotted line with arrow for u6" for "solid line with arrow"

p 107 Figure 4.5: "dotted line with arrow" for "solid line with arrow"

p 107 Figure 4.5: "member unit 1" for "member unit 2" related to u1

p 111 Figure 4.6: "member unit 1" for "member unit 2" related to u1

p 116 first line: "an" for "a"

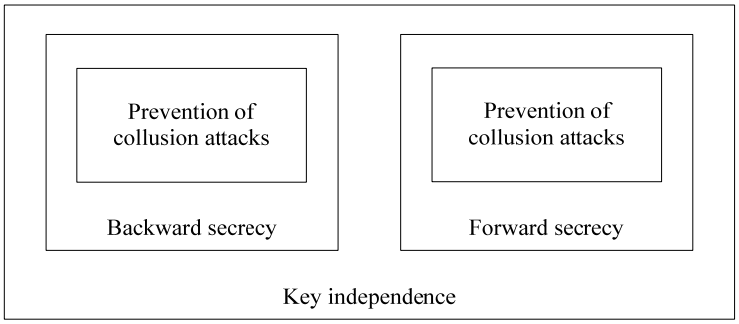
p 178 line 4: "7" for "5"

p 216 Step 1: "$GTEK_{news}$" for "$ATEK_{news}$"

ADDENDUM

p 16: Add at the end of para 2:
“secure group key management for wireless cellular networks [Um & Delp, 2006].

p 76: Delete Figure 3.3 and add new Figure 3.3



p 76: line 3: delete “ ‘Forward secrecy’ subsumes ” and read “ ‘Forward secrecy’ and ‘Backward secrecy’ subsume ”

p 76: line 9: delete “the forward secrecy” and read “the forward secrecy and backward secrecy”

p 76: line 10: delete “the current group key” and read “the group key which they are not entitled to know”

p 110: line 14: delete “on behalf of the CKC” and read “by multicast.”

p 121: line 10: delete “the CTEK”

p 130: Add a new paragraph at the end of para 1:

“The reliable message delivery mechanism can detect misbehaving leaders to ensure the delivery of keying materials. If a leader does not function normally, all the resending requests are sent to CKC. Based on the number of received resending requests from a certain unit, CKC can detect the malfunction of a leader and replace it following the procedure of leave operation.”

p 133: Add a footnote: “the security analysis of HGKM is presented in Chapter 6.”

p 133: Add a paragraph at the end of para 1:

“The wireless network data rate and maximum transmission unit (MTU) are also required to be considered when designing group key management systems. Currently,

the data rate of wireless network is just several hundred kilobits. It requires the rekeying materials to be as small as possible to reduce the communication cost in wireless networks. Along with the rapid development of wireless networks, the data rate can be increased to dozens megabits or even more. Therefore, in wireless broadband networks, the data rate is not the obstacle to the implementation of group key management systems. The wireless MTU is the maximum package that can be transmitted over wireless networks. Generally it is 1500 kilobytes, the same size as the MUT in Ethernet. Before transmission, rekeying materials may be divided into several transmission units if they are bigger than one MTU. Because wireless communication is an unreliable transmission, the whole rekeying message is required to be retransmitted if one transmission unit is not received by group members. Therefore, it is an advantage to put all rekeying materials into one MTU to reduce the communication cost. As described in section 4.3, HGKM adopts a distributed structure, operation unit, to manage group members. The number of new keys is determined by the number of members in the operation unit. When the size of the operation unit is determined, the number of new keys for rekeying is also settled. Due to the small size of the operation unit, all the new keys can be put into one MTU for transmission. In this thesis, we assume the size of the operation unit is 32, thus the number of new keys is 16 and 15 for the join and leave actions respectively. If we apply the Advanced Encryption Standard (AES) with 160-bit key size, the sizes of the new keys are 320 kilobytes and 300 kilobytes. We can see that all the rekeying materials can be put into one wireless MTU. Moreover, based on the analysis in section 4.3, the number of new keys for the join and leave actions is only related to the size of the operation unit and does not change when the size of the group increases. This is an advantage of HGKM to reduce the communication cost for a large and dynamic group.

p 184: Add a footnote: “the security analysis of MOKM is presented in Chapter 6.”

p 239: Add the following paragraphs after the first para:

“Figures 6.11 and 6.12 illustrate the communication cost of the join and leave actions

for CWGKM and TMKM. In these examples, without loss the generality, we assume the size of the operation unit is 32, the degree, α, is 2 and the level, *L*, is 8.

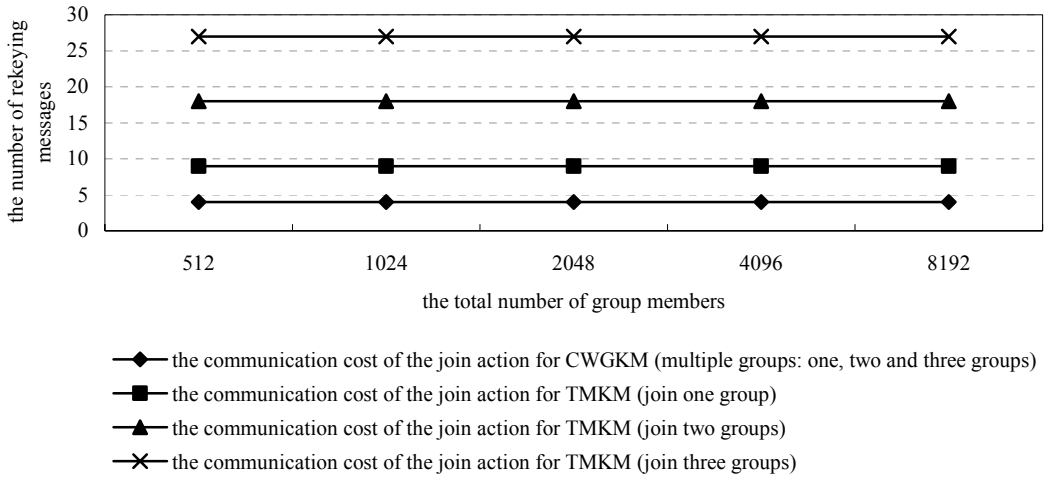


Figure 6.11 The communication cost of the join action for CWGKM and TMKM

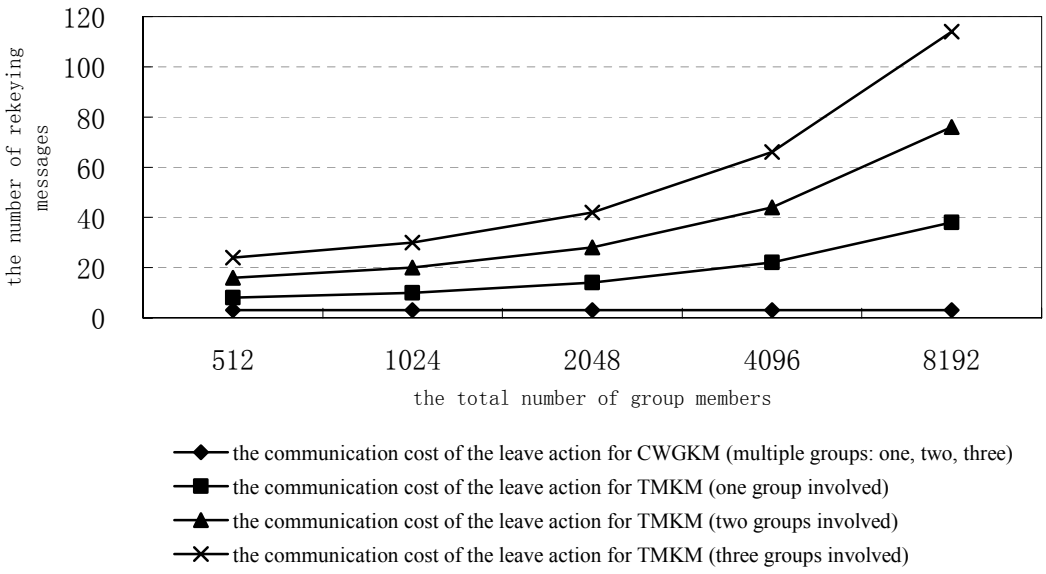


Figure 6.12 The communication cost of the leave action for CWGKM and TMKM

From Figures 6.11 and 6.12, it can be noticed that the communication cost of TMKM is higher than that of CWGKM for both the join and leave actions. Due to the application of micro-key and membership-oriented management, the communication cost of CWGKM is a constant value and immune to the change of the group size and the growth of the number of groups involved in the membership changes. On the other hand, TMKM has the constant communication cost for the join action because the size of the level, *L*, is fixed during the life circle of group applications. For the

leave action, the communication cost of TMKM increases with the change of the group size. It is because the number of the members attached to a node increases with the growth of the group size. In the scenario of multiple-membership changes, the increase of the communication cost of TMKM is linearly proportional to the number of the groups involved in the membership changes.

Figures 6.13, 6.14 and 6.15 illustrate the computation cost of the join and leave actions for CWGKM and TMKM.

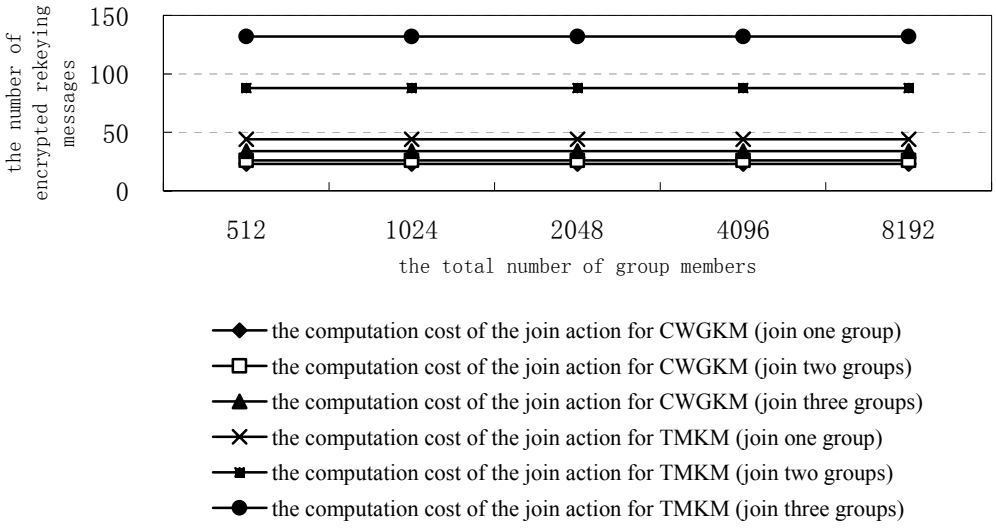


Figure 6.13 The computation cost of the join action for CWGKM and TMKM

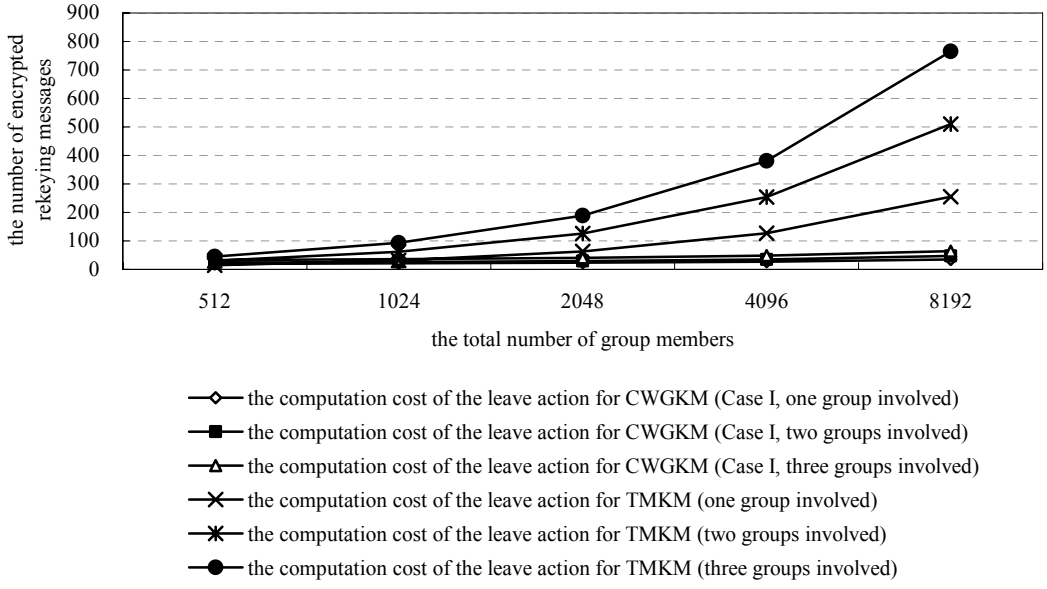


Figure 6.14 The computation cost of the leave action for CWGKM (Case I) and TMKM

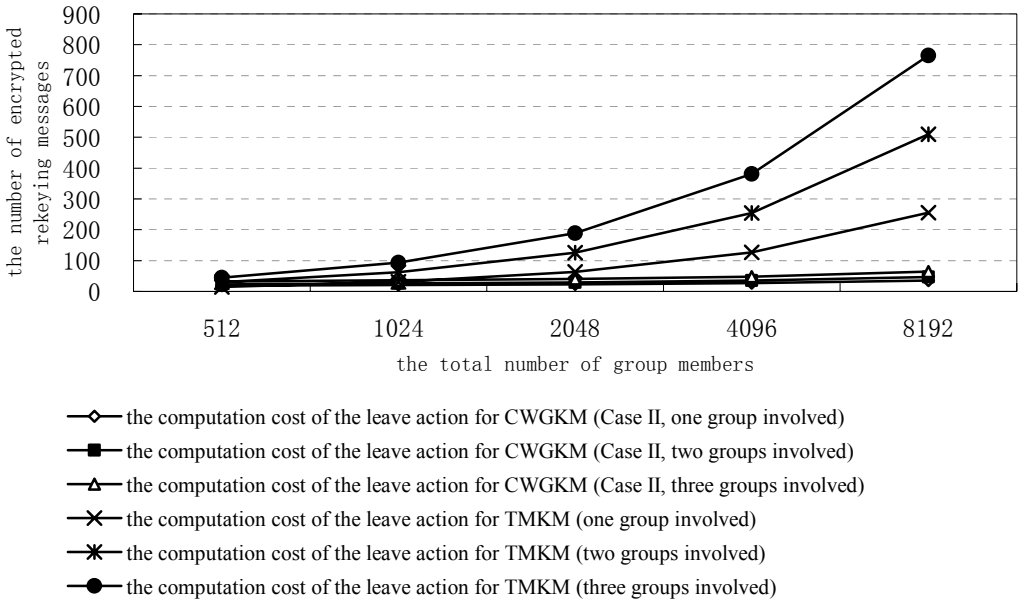


Figure 6.15 The computation cost of the leave action for CWGKM (Case II) and TMKM

From Figures 6.13, 6.14 and 6.15, we can see that the results are similar to that of the communication cost. Due to the same reasons, CWGKM has less computation cost than that of TMKM. The computation cost of the join and leave actions for CWGKM increases slightly with the growth of the group size and the number of the groups involved in the membership changes. On the other hand, the computation cost of TMKM increases sharply with the growth of the group size. Moreover, in the situation of multiple-membership changes, the computation cost of TMKM increases linearly with the growth of the number of the groups involved in the membership changes.

p 267: Add a new reference:

[Um & Delp, 2006] Um, H., and Delp, E. J. (2006). *A Secure Group Key Management Scheme for Wireless Cellular Networks*. In Proceedings of the Third International Conference on Information Technology: New Generations (ITNG06), pp. 414 - 419