A Key Management Framework for Secure Group Communication in Wireless Mobile Environments

Miss Laiha Mat Kiah

Technical Report
RHUL–MA–2008–18
May 2007

Department of Mathematics
Royal Holloway, University of London
Egham, Surrey TW20 0EX, England
http://www.rhul.ac.uk/mathematics/techreports

A Key Management Framework for Secure Group Communication in Wireless Mobile Environments

Miss Laiha Mat Kiah

Thesis submitted to the University of London
for the degree of Doctor of Philosophy

Information Security Group
Department of Mathematics
Royal Holloway, University of London

2007

Declaration

The work presented in this thesis is the result of original research carried out by myself, in collaboration with my supervisor, whilst enrolled in the Department of Mathematics as a candidate for the degree of Doctor of Philosophy. This work has not been submitted for any other degree or award in any other university or educational establishment.

Miss Laiha Mat Kiah
May, 2007

Acknowledgements

The past four years have been thus far the most challenging, interesting, and rewarding part of my life. I am very thankful and grateful that I have crossed paths with many wonderful people who have helped me in many ways in my pursuit of a PhD at Royal Holloway, University of London.

First and foremost, I would like to thank my supervisor, Dr Keith M. Martin, for his excellent supervision, teaching, guidance, patience and support in this research, and for getting me through the unlimited and vast world of thoughts towards completing my PhD degree.

I would also like to thank my academic advisor, Prof. Peter Wild, for his constructive comments. Special thanks go to my family, friends and all ISG members in the department, who have helped and supported me over these years.

Above all, I want to thank God for His grace that makes it possible for me to produce this work, and complete my PhD.

To my husband *Nor Asrul Mohd Noor*,
and
my son *Danial Irfan Nor Asrul*,
for their unconditional loves and supports.

Abstract

Multicast functionality can be used to enable group communication more efficiently than the traditional unicast networks. Like unicast environments, multicast or group-based applications are expected to deliver same level of service to both end users and service or content providers. One of the problem areas concerns with provision of secure group communication is the management of keying material, which is primarily managed by an infrastructure, referred to as a group key management framework (GKMF). The main function of a GKMF is providing common cryptographic key(s) to all group members of a multicast group communication.

While security issues pertaining to deployment of secure group communication in fixed unicast networks are widely research, very little consideration is given for establishing such communications in wireless mobile environments (WMobEs). Inherent characteristics of WMobEs such as restricted capabilities of mobile devices, as well as mobility of group members provide further challenge for deploying secure group communication in such environments.

Thus, this thesis concerns key management frameworks for secure group communication in WMobEs.

There are three main parts to the work. First, we begin with an introduction to multicast technology, including its capability to enable group (or multicast) communication. Second, we focus the work on one area, the management of group keying material within a GKMF, including its main components and processes (or protocols). Third, we propose a specification for a GKMF for secure group communication, based on a specific wireless mobile architecture. Finally, we conclude our work by identifying future research directions.

The main contribution of this thesis is to design, specify and analyze a GKMF for group communication in WMobEs.

Contents

List of Figures

List of Tables

Abbreviations

AES:	Advanced Encryption Standard
CPU:	Central Processing Unit
CSCW:	Computer-Supported Cooperative Work
GKMF:	Group Key Management Framework
GSEC:	The Group Security Research Group
GSM:	Global System for Mobile Communications
hex:	hexadecimal
IETF:	Internet Engineering Task Force
IGMP:	Internet Group Management Protocol
IP:	Internet Protocol
IPv4:	IP version 4
IPv6:	IP version 6
IRC:	Internet Relay Chat
IRTF:	Internet Research Task Force
IS-41:	Interim Standard-41
Kbps:	Kilobit per second
KDC:	Key Distribution Centre
KTC:	Key Translation Centre
MAC address:	Mandatory Access Control address
MAC:	Message Authentication Code
mAH:	Milliamp Hour
MANETs:	Mobile Ad-hoc Networks
MB:	Mega Byte
MBONE:	Multicast Bone
MHz:	Megahertz
MSEC:	Multicast Security
PDA:	Personal Digital Assistant
PIN:	Personal Identification Number
PPV:	Pay per view
RAM:	Read Access Memory
SA:	Security Association
SMuG:	Secure Multicast Research Group
SOHO:	Small Office Home Office
TV:	Television
UMTS:	Universal Mobile Telecommunication System

WG:	Working Group
WLAN:	Wireless Local Network
WMobE:	Wireless Mobile Environment
WWW:	World Wide Web

Chapter 1

Introduction

This chapter introduces the direction of our work, the motivation that drives us into carrying out this research, and the contributions of the work.

In *Section 1.1*, we introduce our work and identify the motivation for this research. In *Section 1.3*, we present our main research contributions. Finally, in *Section 1.4*, we briefly outline the main structure of the thesis.

1.1 Research Motivation

In recent years, group-based applications have significantly grown in popularity. These include:

- Multimedia conferencing, such as video, audio or tele-conferencing.
- Information dissemination services (often referred to as *push technologies*), such as stock quotes, special news or software updates.
- Satellite TV distribution services, such as Pay Per View (PPV) channels.
- Online interactive forums, such as virtual classrooms.

As a result, there has been a rejuvenation in the interest shown in *multicast* technology first proposed by Deering (Deering, 1989) in late 1980s. This multicast technology exhibits special functionality which enables efficient deployment of group communication, where a message can be sent from a single sender to multiple receivers in one go (Deering, 1989), (Almeroth, 2000), (Hardjono and Tsudik, 2000).

In this thesis we are particularly concerned with the provision of security for group communication. According to the Internet Engineering Task Force (IETF), one of the three main problem areas in secure multicast is concerned with the management of cryptographic keying material, which includes the generation, distribution and updating (or re-keying) of keys (MSEC, 2007). As a result, there has been considerable research activity in this area.

A vital component of any security architecture for group communication based on cryptographic services is the design of a group key management framework (GKMF). A GKMF specifies the entities and processes involved in governing all aspects of the management of cryptographic keys. It is important to ensure that protocols for handling cryptographic keys are secure and uphold the security objectives of particular applications. It is also imperative to protect multicast group applications from security threats such as eavesdropping on confidential traffic, injecting false data traffic, modifying key values, or masquerading to join a multicast group.

The main security objectives for multicast communications are similar to traditional (unicast) communications, namely the provision of *confidentiality*, *integrity* and *authentication* security services. These are intended to ensure that confidential information remains unexposed, information received by the group members is correct and has not been modified during transit, and that communicating entities (including the group members of a multicast group) are the ones who they claim to be. A GKMF provides all the management tasks to maintain and protect the keys required to implement all of these services.

The first GKMF for secure multicast communication was proposed as part of the Internet Standards pertaining to key management (Harney and Muckenhirn, 1997). Subsequent research on GKMFs has attempted to improve on this framework (Mittra, 1997), (Wong et al., 1998), (Waldvogel et al., 1999), (Hardjono et al., 2000a), (Hardjono et al., 2000b) and (Baugher et al., 2003). However, all of these GKMFs are intended for deployment in wired environments (such as the original Internet).

However, very little consideration has been given to developing GKMFs for wireless mobile environments (WMobEs). Such environments place additional constraints on the design of a GKMF since, for example, many devices that operate in wireless environments have low computing power and storage capacity (Forman and Zahorjan, 1994) and (Chlamtac and Redi, 1998). Furthermore, hosts that are wireless and mobile aggravate the complexity of designing a secure GKMF since, for example, group members from one area may be allowed to move to another area while still remaining in a group session.

Currently, there is no detailed specification of a GKMF for WMobEs. The main contribution of this thesis is to design, specify and analyze a GKMF for group communication in WMobEs.

1.2 Research Objectives

The aim of this thesis is to design, specify and analyze a group key management framework for secure group communication in wireless mobile environments.

The main research objectives of this thesis are, as follows:

- To provide backgrounds on multicast technology and its capability to enable group communication.
- To identify different security challenges for establishing secure group communication in WMobEs.
- To establish a generic GKMF, which includes main components and processes (protocols) if a GKMF for secure group communication in WMobEs is to be specified and designed.
- To design a specification of GKMF for secure group communication in WMobEs.

- To provide a basic analysis of the proposed GKMF to see to what extent that the framework meets its design requirements and its security objectives.

1.3 Research Contributions

This thesis provides a specification of a GKMF for a WMobE. We believe that our work makes the following contributions to knowledge of secure group communications in mobile environments:

(1) We provide a critical analysis of the different challenges of secure multicast between wired and wireless mobile environments.

(2) We establish a generic group key management framework, which describes the essential and desirable components that need to be addressed if a group key management framework for wireless mobile environments is to be specified.

(3) We design a specific group key management framework for wireless mobile environments. This includes a specification of the main group key management protocols.

(4) We conduct a basic analysis of the proposed framework, in terms of functionality, security and performance.

1.4 Thesis Structure

The rest of the thesis is organized as follows:

Chapter 2 Introduction to Multicast and Group Communication... We introduce multicast and group communication.

This includes defining the terms that we use throughout the thesis.

Chapter 3 Security in Group Communication... We discuss the security (research) issues pertaining to multicast group communication, and give an overview of activities and standards for establishing secure group communication.

Chapter 4 Group Key Management Frameworks (GKMF)... We discuss the provision of key management for group communication; referred to as a *Group Key Management Framework (GKMF)*.

Chapter 5 GKMF: Design Challenges in Wireless Mobile Environments (WMobEs)... We identify specific issues that pose further challenge for establishing GKMFs in wireless mobile environments (WMobEs).

Chapter 6 A Generic GKMF Model for Wireless Mobile Environments... Here, we propose a generic model of a GKMF for WMobEs.

Chapter 7 Existing GKMFs... We look at existing GKMFs to show to what extent they fit into our generic model proposed in Chapter 6.

Chapter 8 GKMF for WMobE: Scope and Requirements... Based on the blueprint of the generic model, we specify in detail our GKMF proposal for WMobEs. This chapter presents the scope and requirements of our framework.

Chapter 9 GKMF for WMobE: GKMF Protocols... We describe the design of protocols for our GKMF.

Chapter 10 GKMF: Analysis of the Proposal... We provide a basic analysis of the proposed framework. This includes analysis each of protocol, and general analysis with regard to performance and scalability of the proposed GKMF.

Chapter 11 Conclusions and Future Work... We conclude our work, and provide suggestions for future research.

Chapter 2

Introduction to Multicast and Group Communication

This chapter gives an introduction to multicast (including definition of terms) and its relationship to group communication.

In *Section 2.1*, we introduce the main terms and definitions that we will use throughout the thesis. In *Section 2.2*, we look at the environments within which multicast group communications operate. Finally, in *Section 2.3*, we identify different types of multicast application.

2.1 Terms and Definitions

Since the successful testing of Audiocast at the 1992 Internet Engineering Task Force (IETF) meeting in San Diego, U.S.A (Casner and Deering, 1992), the idea of incorporating multicast features in network products has been highly attractive to many vendors. Progress has been slow since initial deployment, especially compared to the likes of the World Wide Web (WWW) and the Hypertext Transfer Protocol (HTTP). The reason for this is because multicast features require additional intelligence in the network which introduces non-trivial amounts of state and complexity in both core and edge routers (Gong and Shacham, 1995). However, multicast communication is becoming increasingly important and is a subject of great research interest.

Before we proceed further, it is necessary for us to clarify some basic terminology that we will use throughout this thesis. In the following sections, we explain the differences between *multicast*, *broadcast* and *unicast*. We also demonstrate the advantages that multicast offers for enabling group communication.

2.1.1 Unicast vs Multicast

Extrapolating from definitions in (Deering, 1989), (Miller, 1999), (Ammer, 2000), (Wittmann and Zitterbart, 2001), (Goyeneche, 2004) and (Ciscosystem, 2006), the term *multicast* can be defined as,

> *An internetwork function that allows data to be delivered to a specific group of nodes (or recipients) by a single transmission.*

From a data sender point of view, *multicasting* allows data to be sent from the source (which is the *sender*) only once, and the network will make copies of data and transmit the data to multiple destinations (which are known as the *recipients* of the multicast data) which have been determined prior to transmission. This special feature enables data to be efficiently sent to a group of recipients, and is therefore attractive for group-based application services such as video conferencing, as well as data delivery services such as stock quote, news or weather updates. Multicast is much more efficient then traditional *unicast* methods, which only transmit data to one intended recipient.

We illustrate an example of unicast versus multicast data communication in *Figure 2.1.* The left figure shows *unicast* data communication, while the right figure shows *multicast* data communication. From *Figure 2.1*, *unicast* and *multicast* data transmissions are indicated with **bold** arrows. Based on one-to-many relationship (see *Section 2.3*), it shows a single sender (S) sending data to a group of receivers (R).

If group communication is to be accomplished via conventional data communication based on unicast, as shown on the left of *Figure 2.1*, data to all group

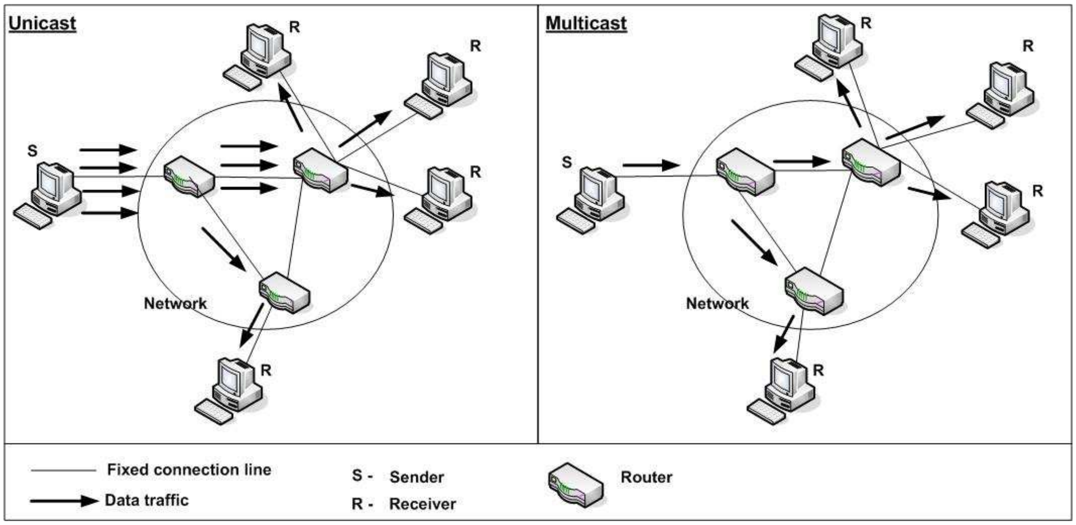


Figure 2.1: An example of unicast vs. multicast communication.

members needs to be sent across the network separately. Thus, in order to transmit to n recipients (group members), n pieces of data need to be transmitted. As illustrated on the right however, multicast allows a sender of data to transmit only one copy of the data to n recipients. This is achieved when a router (with a built-in multicast function at the network level) makes copies of the data and transmits it to the intended recipients via the nearest routers. End routers (closest to the recipients) will then complete the transmission, and send the data to the intended recipients.

2.1.2 Multicast vs Broadcast

A closely related concept to multicast is *broadcast*. Commonly used in radio and TV transmission, broadcast is easily understood as a way to transmit data or messages to all recipient nodes in a network. Both broadcast and multicast allow data to be transmitted to more than one recipient at a time. However, in regards to group communication, multicast offers a better solution than broadcast in several aspects (Gong and Shacham, 1995), (Huitema, 1995), (Reid, 1997), (Almeroth, 2000), (Comer, 2001), (Hardjono and Dondeti, 2003) and (Perrig and Tygar, 2003):

- *Host coverage.* Broadcast includes everyone (all hosts) in the network. This is due to its indiscriminate transmission which can be received by anyone having the correct equipment in place. For example:

  (i) houses with a specific dish can receive satellite channels cast by a satellite station.

  (ii) computers connected to an Ethernet can receive messages cast by the network.

  On the other hand, multicast data is not sent to all hosts, but is rather targeted to a predetermined group of hosts which have specific network addresses.

- *Internet Protocol (IP) context.* Broadcast is usually designated by a single address assigned to all stations in the network (such as an Ethernet packet with a MAC address that can be received by all stations) (Reid, 1997). On the other hand, multicast offers more restricted access with an IP multicast address (Deering, 1989) designating a certain group of hosts in such a way that any transmission from one host in the group is received by all other hosts within the group. More precisely, data packets can only be processed by multicast group members with a correct multicast address. For this to work, a particular address is reserved for that purpose and is found in the destination address field of the message (see *Section 2.2*).

- *From a recipient's context.* With broadcast, all recipients will act upon the received message, whereas with multicast only those recipients that have been configured to respond to the destination address in the message will do so. Thus, multicast saves network resources (such as the CPU time) by allowing only the intended group of recipients to further process the message received.

2.1.3 Group Communication vs Multicast Communication

Throughout this thesis, the terms *group* and *multicast group* carry the same meaning and will be used interchangeably. Informed by various sources such as those in (Deering, 1989), (Forman and Zahorjan, 1994), (Gong and Shacham, 1995), (Chlamtac and Redi, 1998), (Canetti et al., 1999), (Diot et al., 2000), (Hardjono and Tsudik, 2000), (Bruschi and Rosti, 2002), (Hardjono and Dondeti, 2003) and (Baugher et al., 2005), we will avoid formal definitions and interchange the terms *multicast communication*, *group communication* and *multicast group communication*. This approach is also supported by the following sources gathered from the *American Heritage* Dictionary of the English Language as well as from the *Cisco* Internetworking Terms and Acronyms (Ammer, 2000) and (Ciscosystem, 2006), where both terms carry similar meanings, as follows:

> *Multicast communication* (Ciscosystem, 2006). A single communication (like an audio, a video or packets) made and copied by the network and sent across a network to a specific subset of network address.
>
> *Group communication* (Ammer, 2000). A communication that occurs in an assemblage of persons or objects, all interconnected and capable of communicating with each other, in ways that are securely isolated from all other users on the network.

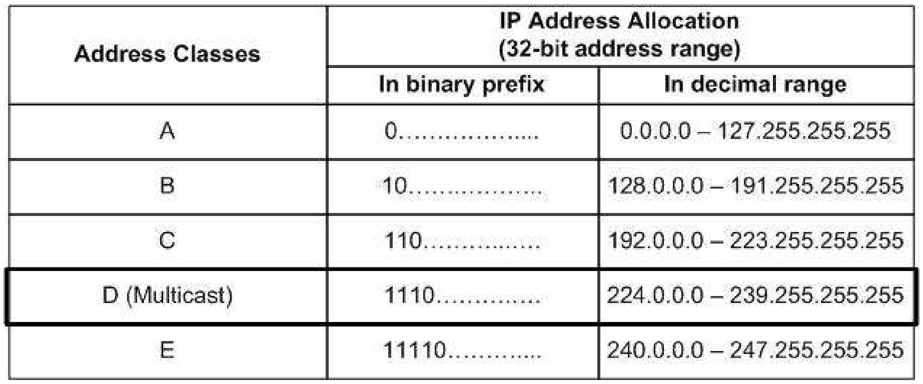
2.2 Multicast Environments

The popularity of multicast has grown considerably with the wide use of the Internet, as well as the increasing demand for group-based applications such as online forums, pay per view channels (PPV), various information dissemination services (such as news, weather, or share prices updates), as well as multimedia conferences including video and audio conferencing.

While many internet applications use the conventional *point-to-point* or unicast transmission, *one-to-multipoint* or one-to-many transmission was limited to local network applications. The emergence of new applications over recent years has seen changes in the earlier trend of unicast transmission seen as being inadequate to support group-based applications. Thus, multicast transmission is becoming more popular as the demand for these new group applications increases.

The original protocol that allowed the transmission of data on the Internet is the *Internet Protocol* (IP), which supports unicast communication. For multicast to function, an extension was designed to the original IP architecture to incorporate the interesting features of multicast. This extension is known as *IP multicast*. IP multicast is defined as a transmission of an IP datagram to a group of hosts, identified by a single IP destination address (Deering, 1989). This address must be one of the special addresses designated for the purpose of multicast communication.

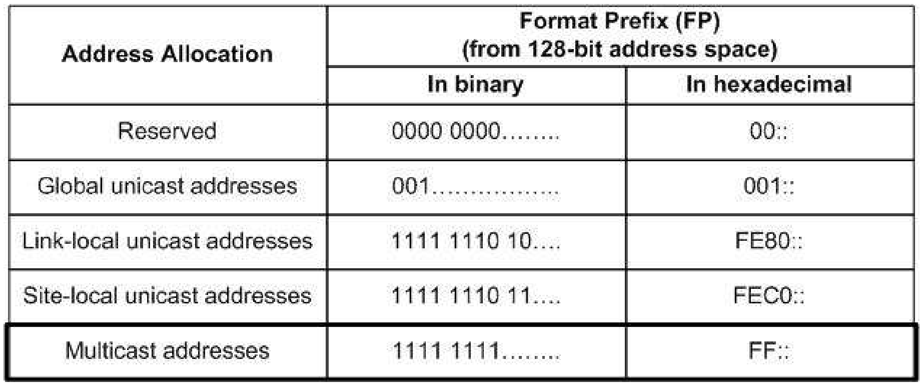
Based on the sources in (Deering, 1989), (Davies, 2003), (IANA, 2005) and (Hinden and Deering, 2006), we illustrate the allocation of IP multicast addresses in the form of two tables. *Table 2.1* gives the allocation of multicast addresses in IPv4 (IP version 4), while *Table 2.2* gives the IPv6 (IP version 6) version of multicast address allocation. While an IP multicast group is identified by a class D IPv4 address which ranges from 224.0.0.0 to 239.255.255.255, in IPv6 multicast addresses always begin with 1111 1111 (binary), or FF (hex). This set of addresses can be used for defining different multicast groups (except for several designated addresses which are reserved and will never be used).



Address Classes	IP Address Allocation (32-bit address range)	
	In binary prefix	In decimal range
A	0…………….....	0.0.0.0 – 127.255.255.255
B	10……………..	128.0.0.0 – 191.255.255.255
C	110…………..….	192.0.0.0 – 223.255.255.255
D (Multicast)	1110………….	224.0.0.0 – 239.255.255.255
E	11110………....	240.0.0.0 – 247.255.255.255

Table 2.1: Multicast address range in IPv4.

Just like conventional communications that are based on unicast, IP multicast is an unreliable, unordered, and best-effort datagram service. In other words, it does not guarantee that a datagram will arrive at all the destination group nodes, and it is possible that the order of receiving the datagram at the end



Address Allocation	Format Prefix (FP) (from 128-bit address space)	
	In binary	In hexadecimal
Reserved	0000 0000........	00::
Global unicast addresses	001.................	001::
Link-local unicast addresses	1111 1110 10....	FE80::
Site-local unicast addresses	1111 1110 11....	FEC0::
Multicast addresses	1111 1111........	FF::

Table 2.2: Multicast address prefix in IPv6.

nodes may change during transmission.

In this early age of multicast technology, when multicast functionality has not yet been widely integrated into traditional network routers, an experimental multicast network called *multicast bone* (MBONE) (Savetz et al., 1998), (Goyeneche, 2004), (Devereaux-Weber, 2006) was constructed as a test bed to realize multicast communication. The original purpose was to carry out the first audiocast of IETF meetings. MBONE is a virtual network consisting of IP tunnels (with virtual point-to-point links) between multicast routers that join the multicast network together. As a consequence, MBONE's main functional aspect was providing an environment for multimedia conferences for various groups of people from across the Internet.

These tunnels are particularly useful in many conventional unicast networks where many network routers are not multicast capable. For this to work, both end routers (nearest to a group of multicast hosts) need to be multicast capable routers, and IP packets that are addressed to a multicast group are tunneled between these multicast routers. In practice, the datagram of a multicast group is encapsulated into another IP datagram by the nearest multicast router, and sent across the network through one (or more) unicast routers as unicast datagrams, which then forward it to the next multicast router of another multicast group.

We illustrate a multicast communication over a unicast network in *Figure 2.2*, with *Figure 2.2(a)* showing a set of multicast hosts wishing to form a group communication over a unicast network and *Figure 2.2(b)* depicting the tunneling of IP datagrams that occurs between these multicast hosts.

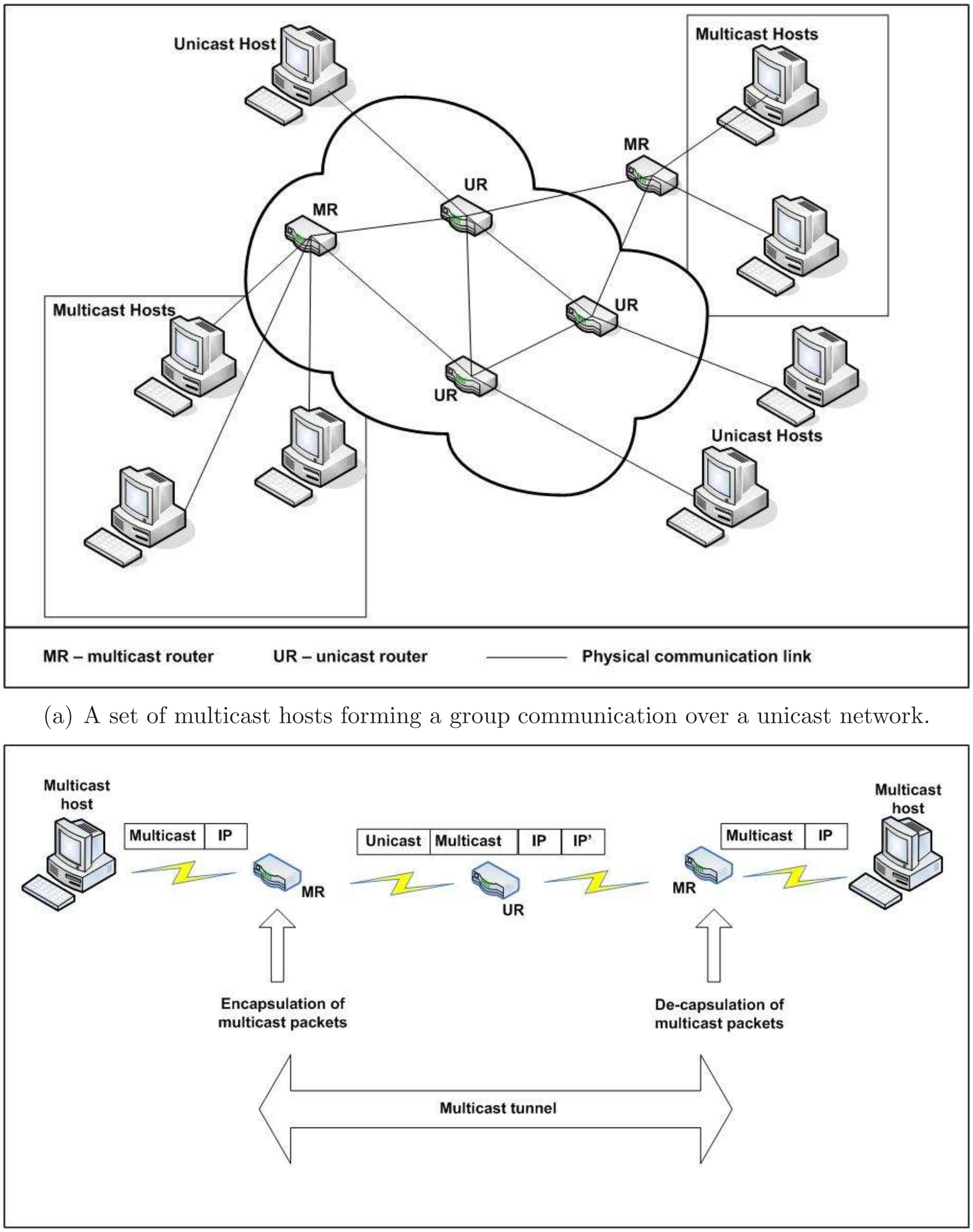
Figure 2.2(a) illustrates communication that occurs between several multicast hosts over a traditional unicast network. This is supported by multicast routers (MR) at both ends of multicast hosts. Here, we show that as this communication occurs over a traditional network, the transmission link may have to go through unicast routers (UR) which normally do not support multicast. One way for this to work is by using a technique called *tunneling* (see *Figure 2.2(b)*).

Figure 2.2(b) illustrates two multicast hosts communicating with one another over a unicast network (as shown in *Figure 2.2(a)*) via a *multicast tunnel*. This is achieved as follows. When multicast IP datagrams leave the host to the nearest multicast router (MR), encapsulation of multicast packets into unicast IP datagrams is done, before transmitting it over a unicast network with unicast router(s) (UR). At the receiving host, when the datagrams reach the multicast router, the same datagrams will go through a de-capsulation process to recover the original multicast IP datagrams, before being transmitted to the host. This process of encapsulation and de-capsulation creates a *multicast tunnel* between multicast hosts for enabling multicast communication over a traditional unicast network.

2.3 Multicast Applications

We have now presented an overview of multicast, including the terms that we will use throughout this thesis. We have also demonstrated that multicast technology is more efficient than broadcast and unicast when enabling group communication.

In this section, we look at different types of multicast application, as well as several instances of group-based applications that could utilize multicast technology. The main types of multicast application can be divided into two



(a) A set of multicast hosts forming a group communication over a unicast network.

(b) Tunneling of IP datagrams that occurs from *(a)*.

Figure 2.2: An illustration of a multicast communication over a unicast network.

categories, which are *one-to-many* and *many-to-many relationships* (Harney and Muckenhirn, 1997), (Hardjono et al., 2000a), (Wittmann and Zitterbart,

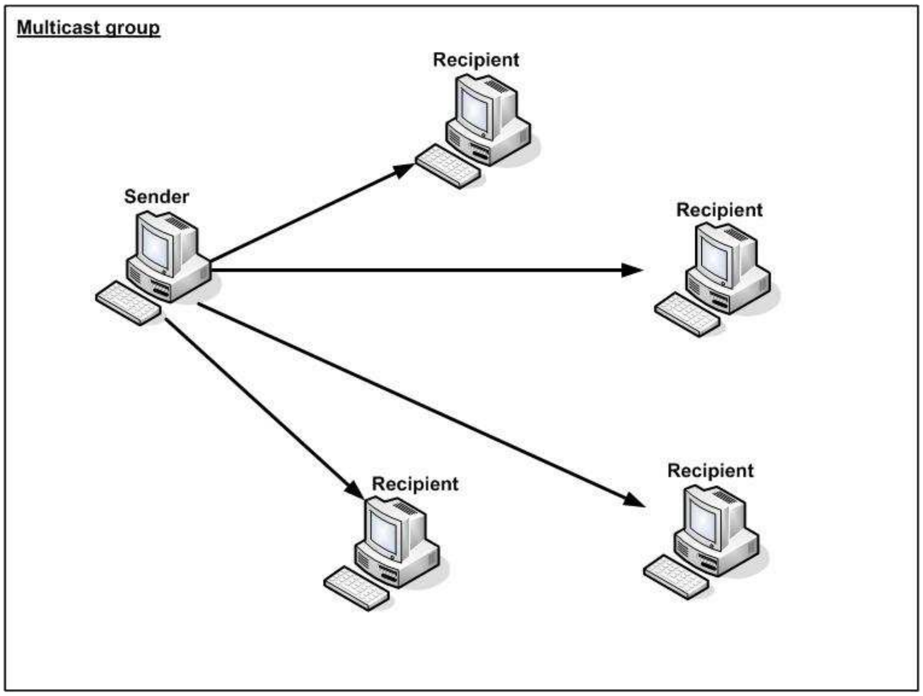


Figure 2.3: A one-to-many multicast (or group-based) communication.

2001). Each of these determines the relationship between the *sender* (S) and the *recipients* (R) of multicast data.

2.3.1 One-to-Many Relationships

One-to-many relationships correspond to *one-Sender* to *many-Recipients*. In this case, there is one entity that is the sender of the group, while one or more entities will be the recipients of the group communication. *Figure 2.3* illustrates this type of relationship within a multicast group, where there is one entity that acts as the sole sender of the multicast data, while the others are the recipients.

Amongst the examples of such group communications are the following:

- *Data distribution.* Push technologies such as stock quote services, weather or news data, updating of sales information or, price list to all branches, as well as advertisement dissemination based on consumer habits or place of residence.

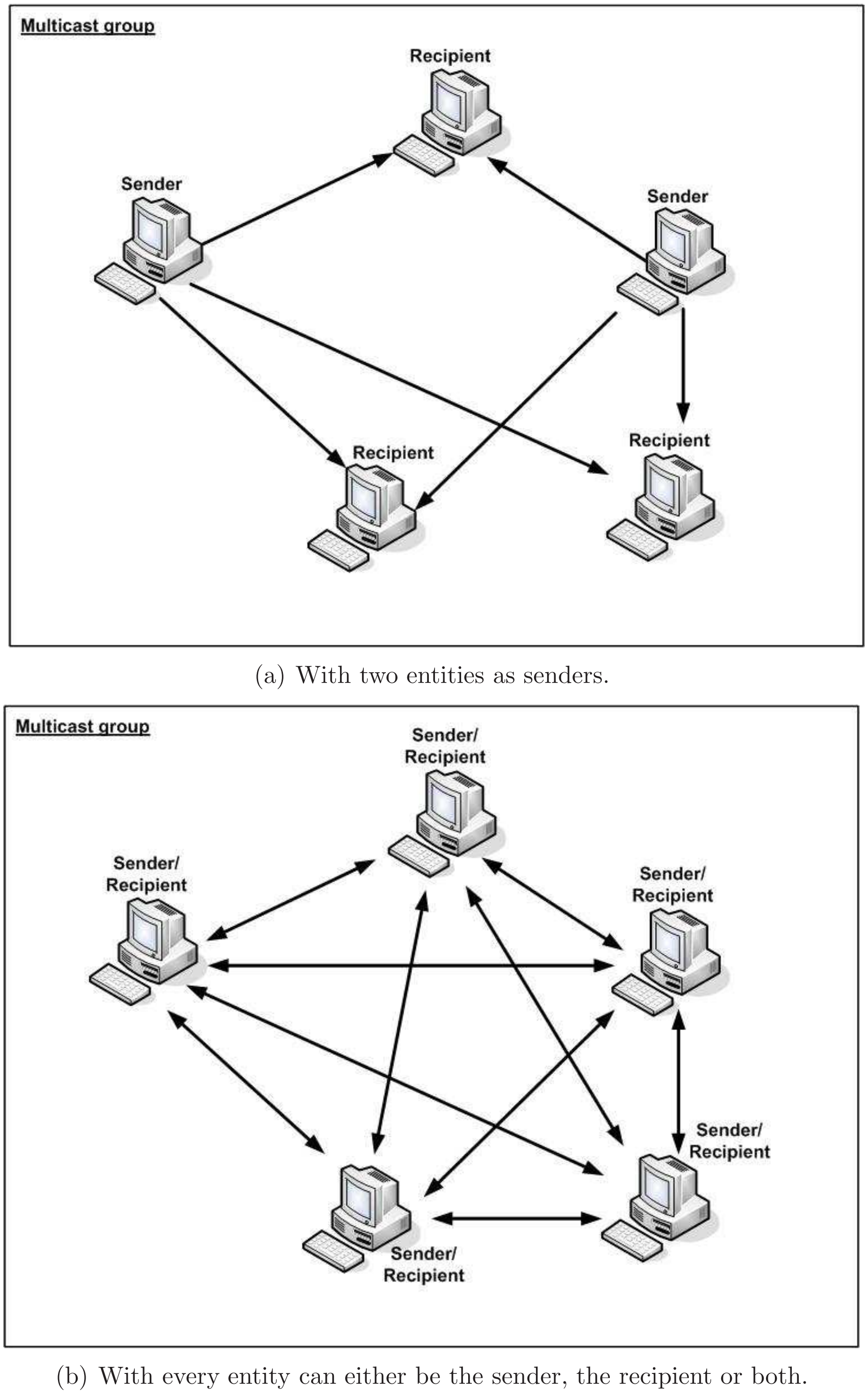
- *Streamed data delivery.* Widely referred to as broadcasting, such as TV broadcasting as well as Pay Per View (PPV) channels.
- *Software distribution.* Software upgrades in a company, as well as the distribution of updates by software manufacturers of their products over the Internet.

2.3.2 Many-to-Many Relationships

In this type of relationship, many-to-many corresponds to *many-Senders* to *many-Recipients*. In this case, there are one or more entities that will be the sender (s) and/or the recipients of the multicast group. Thus, an entity can be a sender, a recipient, or both. *Figure 2.4* illustrates this type of multicast application. *Figure 2.4(a)* shows a multicast group with two entities acting as the senders of data, while *Figure 2.4(b)* illustrates that every entity within a multicast group can potentially be both the sender and the recipient of data.

Amongst the examples of such group communications are:

- *Audio/video and Teleconferencing.* Also referred to as *Computer-Supported Cooperative Work* (CSCW), group members that are located in different sites will share *white board tools* designed for coordination between members.
- *Distance Learning.* For example teleteaching, which refers to a scenario in which a student in one place is able to participate in a class somewhere else. This requires group communication between those teaching and those learning, since questions have to be asked and solutions need to be discussed. Another example is virtual classrooms, where course participants use the Internet to obtain teaching materials, deliver their assignments, as well as receive feedback from course instructors. Similarly,



(a) With two entities as senders.

(b) With every entity can either be the sender, the recipient or both.

Figure 2.4: Many-to-many multicast (or group-based) communications.

the instructor distributes papers to the virtual class and assignments are collected from the course participants.

Note that some of the above group applications already existed before multicast functionality was designed. These *pre*-multicast era applications are called applications that *emulate* group communication. In these applications, very simple group communication mechanisms are implemented in the applications themselves, but not supported by the communication system that underlies them. While this seems to outdate the need for a multicast function, its implementation and performance are very simple. These were literally designed to emulate group communication properties in the application itself. One example is Electronic Mail Systems that allow you to send the same message to groups of recipients, who are normally specified through mailing lists. Other examples are the distribution of news, chatting on the Internet (such as the Internet Relay Chat (IRC)), as well as game servers that allow web users to play games like Backgammon, Chess and Life together on the Internet.

2.4 Summary

The need for multicast functionality has introduced new challenges, particularly with respect to provision of secure environments for such applications. Consequently, the security aspects and security objectives achieved in *unicast* as well as in *broadcast* environments should also be deployed and achieved in multicast environments. We will look at these issues in the next chapter.

Chapter 3

Security in Group Communication

This chapter discusses security in group communication, including the main security issues specific to group communication.

First, in *Section 3.1*, we look at the current activities and standards pertaining to the provision of security in group communication. Then, in *Section 3.2*, we look at the main security issues of multicast group communication.

3.1 Security Activities and Standards

The first reference to multicast can be found in the PhD dissertation by Deering in the late 1980s, which was proposed as an Internet Standard in (Deering, 1989). Since then, multicast has become part of the research area of the *Internet Engineering Task Force* (IETF) (IETF, 2007), in particular proposing security solutions for group-based communication such as those in (Deering, 1989), (Ballardie, 1996), (Harney and Muckenhirn, 1997), (Wallner et al., 1999), (Hardjono et al., 2000a), (Hardjono et al., 2000b), (Baugher et al., 2003), (Hardjono and Weis, 2004) and (Baugher et al., 2005).

As a large open international Internet community, the IETF formed an *Internet Research Task Force* (IRTF) (IRTF, 2007) to become a sister organization of the IETF, one of whose active research groups is on multicast security. This *Secure Multicast Research Group* (SMuG) (SMuG, 2007) was formed to discuss issues related to multicast security, as well as to investigate standards for secure multicast. SMuG's main intentions were to focus on the security problems relating to IP multicast, with the aim of providing common solutions for a variety of applications. The results obtained were presented to the IETF for potential standardization. In order to expand its research scope, SMuG was replaced by *The Group Security Research Group* (GSEC) (GSEC, 2007), which was another short-lived research group of the IRTF.

With GSEC no longer active, the Working Group (WG) that is currently active in research on multicast, as well as group communication security, is known as *Multicast Security* (MSEC) (MSEC, 2007). Established directly under the IETF organisation, MSEC is anticipated to continue the process of standardizing protocols pertaining to security provision of group communications over the global Internet.

Three main problem areas have been defined by MSEC as the central research areas concerning secure multicast by IETF (IETF, 2007). They are:

- *Multicast data handling.* This covers problems pertaining to the treatment of multicast data by the entities involved in the communication, such as encryption algorithms, as well as data integrity techniques.

- *Management of keying material.* This is concerned with the management of all keying material of a multicast group, including distribution and updating (or re-keying) of all cryptographic keys as well as other security parameters related to the keys (such as information on key expiration periods, or key usage).

- *Multicast security policies.* This area is concerned with all aspects of multicast group security policies, including creation, translation and representation of policy of a multicast group. It is important that group policy is managed properly since policies may be expressed in different ways, they may exist at different levels and they may be interpreted differently according to the context in which they are specified and implemented. For example, policy negotiation and translation (if necessary) should be performed as part of a host joining a multicast group. Otherwise, it is meaningless as new members will not be able to participate in the group communication due to incorrect or inappropriate policies.

Published works in these problem areas can be found in (MSEC, 2007) and (GSEC, 2007).

These problem areas are equally important and ought to receive equal treatment in order to accomplish secure group communication. While some genuinely new solutions are required, some of these problems can also be addressed by adapting accomplishments in traditional unicast environments.

Our interest in particular concerns with security issues that are specific to multicast group communication. We look at this in the following section.

3.2 Security Research Issues Specific to Group Communication

In this section, we present the main security issues pertaining to group communication. Like unicast, provision of security services for multicast group communications is concerned with *confidentiality, integrity, authenticity* as well as *availability* of group communications. As in unicast, an adversary may carry out both passive and active attacks against a multicast communication, such as:

- eavesdropping on confidential communication,
- disrupting a group session,
- blocking data transmission,
- injecting false data traffic,
- masquerading to join a group session,
- initiating a bogus group session, or

- colluding members exchanging information in order to gain unauthorized access to data which may contain cryptographic key and other group related information.

Thus, it is crucial to provide a secure data exchange between group members, which includes use of mechanisms or methods to:

(a) Establish the identity of the originator of a message.

(b) Protect transmitted data, including cryptographic keys, from unauthorized disclosure and modification.

(c) Control group members' access to data.

(d) Enable any member to verify the nature of the session in which he participated.

With no regard to the order of importance, we address the main issues concerned with the provision of security for group (multicast) communication in the following sections.

3.2.1 Group Membership Policy

Multicast's capability of sending data only to a specific group of hosts (group members) requires some additional processing to restrict access to the specific group. This processing includes the management of group membership.

The status of group membership of a multicast group is determined by the group membership policy. This is defined during the creation of a multicast group. The policy of group membership can be categorized into two types, as follows (Gong and Shacham, 1995), (Bruschi and Rosti, 2002), (Hardjono and Dondeti, 2003):

- *Static policy.* Often referred to as *closed* membership, group membership with a static policy offers a restricted approach to allowing hosts to take part in a multicast group. In this case, group membership of a multicast group is predetermined prior to the commencement of a group communication and all group members belong to a certain multicast group throughout its lifetime. For example, a video conferencing facility for a global organization might have a predefined group of hosts corresponding to its multiple branch sites. While no increment in the group size is possible (no new members are allowed to join the group), some circumstances need to be considered with regards to members leaving the group:

  (a) Group members may be permitted to leave during a group session.

  (b) Group members may be allowed to re-join (after leaving) the group session, except for cases where group members are deliberately evicted from the group session. (This is a reasonable policy since group members may well leave a group due to a disconnection that is beyond their control.)

- *Dynamic policy.* Also referred to as *open* membership, a dynamic policy allows any hosts to join (or leave) a multicast group at any time throughout the lifetime of the multicast group.

It is essential to define this group membership policy because it determines the entire set of procedures for a particular multicast group communication.

3.2.2 Key Management

Key management for multicast communication is generally more complex than for unicast environments. In a secure environment, presuming that a request to establish a multicast session amongst a group of hosts is granted, a common group key needs to be distributed to each of the group members prior to the start of the group session.

In secure group communication, specific problems for managing the keying material can be divided into two approaches depending on the group membership policy in place, as follows (Caronni et al., 1996), (Mittra, 1997), (Waldvogel et al., 1999), (Hardjono et al., 2000a), (Noubir et al., 2002) and (Hardjono and Dondeti, 2003):

- *Static approach.* Due to its fixed membership policy, the static approach requires almost no change (or update) in keying material throughout the lifetime of a multicast group except for periodic re-keying (see *Section 4.3.2*). This implies that a new multicast group will need to be created to cope with new members joining the multicast group.

- *Dynamic approach.* The dynamic approach, where any hosts can join (and leave) a multicast group at any time, potentially requires that cryptographic keys be updated whenever there is a change in group membership. The precise need for updating the keys is primarily determined by whether the following services are required:

  (a) *Backward secrecy.* This ensures that past communications, including group keys and their related information, are inaccessible to newly joined members. For provision of backward secrecy, group keying material has to be updated whenever a new join to the group occurs.

  (b) *Forward secrecy.* This ensures that future communications remain inaccessible to departed members. For provision of forward secrecy, re-keying of keying material has to occur whenever an existing member leaves the multicast group.

One of the main challenges in key management for group communication is the distribution of the keys needed by group members of a multicast group. Dynamic group membership aggravates the complexity of the protocols which handle the distribution of cryptographic keys. In particular, it is important to ensure that each group member gets keys for the right group sessions. Additionally, if backward or forward secrecy is required, it is necessary to deny some group members access to specific cryptographic keys.

3.2.3 Group Security and Authentication

In secure multicast environments, as mentioned in *Section 3.2*, provision of security services such as entity authentication, data confidentiality and data integrity are required. However, secure multicast group communication has some specific requirements in these areas (Gong and Shacham, 1995), (Canetti et al., 1999), (Hardjono and Tsudik, 2000), (Almeroth, 2000), (Pessi, 2003) and (Hardjono and Dondeti, 2003):

(a) As hosts may wish to join specific groups, and different groups may have their own security requirements (for example, concerning who can join), it is imperative that:

- Group managers verify that the service provided by a multicast group is accessible only to authorized group members.
- Group members verify that the service they participate in is provided by a genuine source.
- Both (group managers and group members) verify each other's identities.

(b) Different policies (static or dynamic) may require different needs for managing group keys (due to *joins* and *leaves*). In a dynamic policy, if backward and forward secrecy are required then re-keying of group keys will have to occur whenever there is a change in group membership.

3.2.4 Scalability

In general, the term *scalability* refers to the ability of a framework (or mechanisms within a framework) to be extended to cover a larger group of hosts over a wider physical region without too much delay and deterioration in the level of service provided.

In the context of secure group communication, the need for scalability primarily affects the management of keying material. In particular, it affects the choice of types of cryptographic key that are needed for group communication, and the methods by which the keys are updated, keeping in mind that in dynamic environments the size of the group membership may gradually change over time.

While the problem of hosts joining seems to be straightforward (if the provision of backward secrecy is not necessary) and distribution of new cryptographic keys can be supported by the old cryptographic keys, group members leaving poses a much more difficult scalability problem. If the provision of forward secrecy is necessary, new keying material must be sent to the remaining group members in a way that excludes the leaving member. One method that can be used is to send the key updates to each group member separately (each of which is protected by an individual key). This creates a scalability problem if the group is large and/or has a very dynamic group membership (Mittra, 1997), (Wong et al., 1998), (Waldvogel et al., 1999), (Rodeh et al., 2000), (Setia et al., 2000), (Setia et al., 2002) and (Noubir et al., 2002).

Thus, the scalability issues for secure group communication need to be addressed from an early design stage of any key management framework.

3.3 Summary

In this chapter we have discussed security issues in group communication and identified several issues specific to this type of environment. Our main interest is in the management of keying material, in particular the distribution and updating of cryptographic keys, which is crucial to ensure the security of any multicast group communication.

In the next chapter we look at the provision of a key management framework for group communication.

Chapter 4

Group Key Management Frameworks (GKMF)

This chapter discusses group key management frameworks (GKMF) for multicast group communications.

In *Section 4.1* we introduce GKMFs. In *Section 4.2* we discuss methods that can be used to design a GKMF. *Section 4.3* describes the main components of a GKMF. *Section 4.4* discusses security threats that could compromise the multicast group communication security. In *Section 4.5* we describe the main GKMF security requirements for group communication. *Section 4.6* discusses general aspects of key management. Finally, in *Section 4.7*, we present the important features necessary for a good GKMF design. Part of the work in this chapter has been published in (MatKiah and Martin, 2005).

4.1 Introduction

As discussed in Chapter 3, the IETF multicast research group (MSEC) defined three main problem areas pertaining to multicast group communication security. One of these was key management.

A *group key management framework* (GKMF) is an infrastructure comprising the basic entities and functions necessary to provide common cryptographic key(s) to all the members. In particular a GKMF specifies:

- *Entities and relationships.* The placement of entities involved in group communications and their relationships, as well as determining roles and responsibilities for managing the cryptographic keys necessary for multicast groups.

- *Key management processes.* The management operations necessary to control the cryptographic keys that are needed for group members to engage securely in group communications. This includes generation, distribution, as well as updating (or re-keying) of keys. These are expressed in terms of the protocols required to support these key management processes. These include protocols for creating multicast groups, registration of group members to multicast groups, as well as distribution of keys to group members.

4.2 Design Approaches

Design approaches for establishing a GKMF can be classified in a number of different ways. One way is to distinguish between *static* and *dynamic* key management (see *Section 3.2.2*). Static approaches clearly have limited application and more commonly dynamic approaches are required. Group key management frameworks can be further distinguished by two design approaches, depending on whether a designated central entity can be relied upon for key management purposes:

(i) **Centralized schemes**
Centralized schemes, including those in (Ballardie, 1996), (Wong et al., 1998), (McDaniel et al., 1999), (Hardjono et al., 2000a) and (Baugher et al., 2003), require a central entity to govern and manage group keying material. As the main point of security reference, all group members are required to trust this entity. The advantages of adopting centralized schemes are that:

(a) They are easy to manage, since the provision of trust is focused on one entity.

(b) They save some transmission overheads, since authentication of a central entity (such as a group or key manager) may only need to be done once by group members during a multicast group session.

On the other hand, centralized schemes share inherent drawbacks as follows:

(a) Bottlenecks may occur if there is implosion of transmissions where group members send messages to the central entity at the same time, and vice versa.

(b) Having a central entity as the only point of reference creates a single point of failure. If the central entity fails then the whole system collapses, which then results in paralysis of the multicast groups in place.

(c) A central entity requires a large capacity for storing keying materials for the entire system.

(ii) **Distributed schemes**
Distributed schemes, such as those in (Harney and Muckenhirn, 1997), (Steiner et al., 1998) and (Waldvogel et al., 1999) avoid the need for a central entity. In these schemes, each member of a multicast group is equally trusted, and all (or a few) members are required to take part in managing the keying material, including generation of cryptographic keys. For example, one method that can be used is that group members who join early generate the keys and then distribute them to others who join the group later.

The advantage of distributed schemes is that they offer more flexibility. On the other hand, drawbacks of these schemes are that:

(a) They do not always scale well, since distribution of management tasks across larger multicast groups can be complex.

(b) In large networks, the messages exchanged between group members can be prohibitively large.

(c) There is always a risk that colluding members may exchange security information.

(iii) **Hybrid schemes**

Hybrid schemes, such those in (Mittra, 1997) and (Hardjono et al., 2000b) are a combination of the two earlier approaches (centralized and distributed). These schemes are based on a distributed hierarchy of trusted entities for key management purposes. For example, in a two-level hierarchy one or more entities are responsible (at the first-level) for managing sub-entities (at the second-level). A sub-entity at the second-level may govern other lower-level entities. From a bottom-up view, lower entities are dependent on the higher-level entities.

These schemes potentially share both the advantages and disadvantages exhibited in centralized and distributed schemes. Since the properties can be fine-tuned using varying levels of hierarchy, the hybrid approach is quite attractive for designing a GKMF.

4.3 Main Components

In this section we look at the main elements that form a GKMF. As mentioned in *Section 4.1*, these fall into two categories:

4.3.1 Entities and Relationships

The main entities involved in a GKMF are:

- *Group members.* Group members consist of at least one *sender* (who sends the data) and at least one *recipient* (who receives the data).
- *Group manager(s).* Often referred to as a *group controller*, *key server* or *key manager*, a *group manager* (GM) controls all group processes, such as registration of group members to a multicast group. In particular, the GM manages the cryptographic keys that are needed for group communication, including the generation and distribution of such keys to group members.

Note that a group manager's role may be performed by separate entities, one of which is responsible for all general activities that concern a multicast group, such as group membership policy, while the other is primarily concerned with security aspects such as group key management.

4.3.2 Key Management Processes

The essential processes identified within a GKMF are described as follows:

- **Formation of groups**. Formation of a multicast group can be further divided into two processes:

  (a) *Creation of multicast groups*
  At the network level, creation of a multicast group can be done by a host sending a request to a network using the *Internet Group Management Protocol* (IGMP) (Deering, 1989) and (Williamson, 2000). In return, the network kernel assigns a specific multicast address for the group (see *Section 2.2*). At this point, all the information related to a multicast group such as group membership policy, as well as the cryptographic keys needed for a group communication, is determined.

  (b) *Initial registration of group members*
  Once the interest to join a particular multicast group is determined, a host instructs the network that he wishes to receive data sent to a specific multicast group (at the application level, this is usually indicated by a host requesting a group service on the Internet). When that happens, it is considered that the host *joins* the group.

  From another perspective, any host who wishes to join a multicast group sends a *join* request to a group manager. Presuming that the host is granted permission to join the group, group related-information, in particular the cryptographic keys needed for group communication, is exchanged between the group manager and the group member.

- **Generation and distribution of cryptographic keys**. Cryptographic keys can be symmetric, asymmetric or a combination of both, depending on the security objectives or preferences of particular multicast applications. Most GKMF proposals such as (Mittra, 1997), (Wong et al., 1998), (Waldvogel et al., 1999), (Hardjono et al., 2000a), (Baugher et al., 2003), (Hardjono and Weis, 2004) and (Baugher et al., 2005) use symmetric keys because symmetric algorithms have lower computational complexity and are faster than asymmetric algorithms (Gove, 2000), (Ikbal, 2003), (Dankers et al., 2004).

  Using symmetric cryptography, the main keys needed for a multicast group communication normally consist of:

  (a) *Individual keys*
  Often referred to as *long-term* keys, an *individual key* is unique for every host (potential group member), and is typically shared with a group manager. Individual keys are generated by a trusted entity in the GKMF (such as a group manager). These keys are usually established prior to the commencement of a multicast group.

  (b) *Group keys*
  Often referred to as *traffic encryption keys* (TEKs), a *group key* is shared by the group members of a multicast group, and is primarily used for securing the actual data communication. Group keys are also generated by a trusted entity such as a group manager. Group keys are usually distributed to every member of a multicast group under the protection of individual keys.

  Where asymmetric cryptography is used, all entities involved in the group communication are assigned asymmetric key pairs (Harney and Muckenhirn, 1997) and (Hardjono et al., 2000b).

  Note that apart from the aforementioned keys, an auxiliary key may be needed for the secure and efficient distribution of a group key to group members of multicast groups (Wong et al., 1998), (Hardjono et al., 2000b), and (Baugher et al., 2003). Thus, instead of having to send the group key separately under the protection of individual keys of group members, it can be sent once via a multicast message protected under the auxiliary key.

- **New member joins**. This process is quite similar to initial registration of group members. Any host who wishes to join a multicast group will need to send a *join* request message to a governing entity such as a GM. If the member is granted permission to join the multicast group then relevant keys need to be delivered to the newly joined member.

  If *backward secrecy* is required then it may be necessary to re-key cryptographic keys whenever a new member joins a multicast group. This will result in all group members including the newly joined member obtaining a new group key.

  Note that new member joins may only be allowed in dynamic policies, since static policies suggest no increment in group members (see *Section 3.2.1*).

- **Existing member leaves**. The process of an existing member leaving requires that any member who wishes to leave a multicast group sends a *leave* request message to a governing entity such as a GM. If *forward secrecy* is needed then re-keying will need to occur in order to update the group with a new set of group keys.

  Unlike members joining, members leaving are considered special because a *leave* can be:

  (a) *Voluntary*

  This type of leave occurs at the request of a group member. A group member may leave a multicast group at any time.

  (b) *Non-voluntary*

  This type of leave is not requested by a group member (for example the ejection of a group member). A managing entity such as a group or a key manager is responsible for managing and initiating non-voluntary leaves.

  Depending on group security requirements, an eviction of a group member (non-voluntary leave) may require re-keying to occur.

- **Re-keying**. The process of re-keying group members with new cryptographic keys may occur due to:

(a) *Group membership change*

Due to new joins (for *backward secrecy*), or due to existing member leaves (for *forward secrecy*). See *Section 3.2.2* for more information.

(b) *Periodic re-keying*

A pre-determined plan to re-key a multicast group after a certain interval (which is often dictated by a group policy, as well as security requirements of a particular application). This is normally determined prior to the creation of a multicast group.

(c) *Expiration of cryptographic keys*

When a key has reached the end of its validity period. Often this type of re-keying is synchronized with the periodic re-keying (as in (b)), whichever occurs first.

(d) *Compromised keys*

When a key used is believed (or suspected) to have been compromised and is no longer considered safe to use.

Re-keying events are normally initiated by governing entities such as group or key managers. When re-keying occurs within a group, new keys will be generated and distributed to all group members.

4.4 Security Threats

From the perspective of key management, threats (active or passive threats, or a combination of both) that may compromise the security of multicast group communications may in particular be targeted at data traffic of a group, which includes messages (the actual data communication), and keying materials (the cryptographic keys and related information).

In this section, we discuss security threats that could potentially compromise multicast group communication security (Stallings, 1999), (Nichols and Lekkas, 2002), (Vines, 2002). They are listed as follows:

- Eavesdropping on group data traffic that contains confidential data or messages as well as keying materials.

- Intercepting data traffic that could result in further malicious use including:
  - (a) Modifying the contents by inserting part or whole new messages (or parameters) into the group data traffic.
  - (b) Keeping the contents for further cryptanalysis.
  - (c) Deleting messages so that the intended member never receives them.
- Recording messages (such as between two targeted group members or key managers) to re-send (replay) at a later time.
- Disrupting or blocking a group session by an adversary (such as flooding the key entities with bogus requests, which could result in denial of service).
- Masquerading as a member to join a multicast group, or to create a bogus session.
- Gaining unauthorized access from exchange of information by colluding members.

It is thus important to address these threats when designing a GKMF.

4.5 Main Security Requirements

We have looked at the main components of a GKMF and identified the key management processes and potential security threats in multicast group communications. In this section, we look at the main security requirements which are specific to multicast group communication.

Based on existing standard definitions ISO 7498-2 (ISO, 1989), ISO/IEC 11770-2 (ISO, 1996b), BS ISO/IEC 9798-1 (BS, 1997) and FIPS PUB 199 (FIPS, 2004), we derive the security and trust requirements identified with a GKMF as follows:

- **Entity authentication**. Both group members and key manager entity(s) need to be able to authenticate and verify each other's identities, and by doing so each entity believes that an entity is who it claims to be. This usually occurs via unicast communication between two entities, such as when a host first contacts the key manager to join a multicast group.

- **Backward and forward Secrecy**. All necessary key materials must be re-keyed whenever there is a change in group membership (either due to new joins or member leaves). While *backward secrecy* controls access to previous communication from the newly joined members, *forward secrecy* controls access to future communication from the leaving members.

- **Data (Message) integrity and authentication**. Both group members and key manager(s) need to be able to check that the data received originates from the claimed entity(s) and that it has not been altered in an unauthorized way. Depending on the level of security of an application, there are two types of checking that can be done, as follows:

  (a) *Group authentication*
  All members within a multicast group need to be able to check that a message received originated within the group, and that the message has not been altered by entities outside the group. This type of data authentication is usually useful in multicast communication that occurs between entities who share a common key, such as the real data communication between members of a particular multicast group.

  (b) *Origin (Source) authentication*
  Each member (group member or key manager entity) needs to be able to corroborate that the source of data (message) received is as claimed. This type of authentication is often important when one entity needs to verify that the security parameters (keys) received are coming from the claimed entity. For instance, a sub-group key manager will need to be able to corroborate that all messages containing the keys originate from the main key manager.

- **Need for trust model**. For secure multicast group communication, we need a trust model that includes:

  (a) An architecture that is compatible with the existing network protocols.

  (b) A framework which is scalable for expansion without affecting too much the level of service and overall system performance.

  (c) A trusted managing entity(s) for managing keying material (in particular generation and distribution of cryptographic keys).

- **Secure key distribution**. All necessary key materials need to be securely distributed to all group members prior to group communication.

- **Secure key updates (re-keying)**. Key updates must be done securely, since a new set of key materials may need to be distributed whenever a key compromise is suspected, the current keys expire, or whenever there is a change in group membership. Group members need to be informed by the managing entity(s) whenever there is a change in the key materials that they are using and when key updates are on the way. The re-keying process should be conducted without disrupting any ongoing communication.

4.6 Aspects of Key Management

Secure and proper handling of all aspects of key management is one of the fundamental requirements of any GKMF design in order to form a secure and trusted model for the deployment of group communication. Based on ISO/IEC 11770-1 (ISO, 1996a), (Stallings, 1999), (Murray, 2000) and (Zou and Thukral, 2006), the main aspects of key management are the provision of the following basic key services:

- *Key generation.* The generation of cryptographic keys for a particular cryptographic algorithm. This needs to be done in a secure and proper manner.

- *Key registration.* The registration of cryptographic keys with entities. Registration of keys is usually done by a trusted registration authority, and usually applied when symmetric cryptographic techniques are used.

- *Key certification.* This applies to public key cryptography, to ensure the association of a public key with an entity. Key certification is provided by a certification authority.

- *Key distribution.* The dissemination of cryptographic keys to the communicating entities. Key distribution can be performed using physical (or manual) techniques, or using a trusted third party such as a key distribution centre (KDC) or a key translation centre (KTC), where keys can be delivered to users by using other keys (often called key encrypting keys).

- *Key installation.* The installation of a key prior to its use.

- *Key update (re-keying).* The ending of the use of one key and beginning of use of another key (see *Section 4.3.2*).

- *Key storage.* The secure storage of cryptographic keys prior to use, for short-term use, or for back-up. For security reasons, keys are usually stored physically in a secure environment, for instance using tamper-resistant hardware. Keys can also be protected by other means, such as by encipherment with other keys, or by controlling access to keys using passwords or PINs.

- *Key derivation.* A special form of key generation, where a key is derived from other keys using some transformation process. It is important to ensure that compromise of the derived key does not reveal the derivation key or other derived keys.

- *Key archiving.* The provision of secure long-term storage for keys. Archived keys may be needed at a later time for generation of new keys or to verify certain claims after the key has expired.

- *Key revocation.* The revocation of a key after key compromise is suspected, or known, or when it has reached its expiration date. Similar to the process of key update, except that there are no subsequent keys involved.

- *Key de-registration.* Part of the key disposal process, a key association with an entity is removed. This is done by a key registration authority.
- *Key disposal.* The disposal or destruction of a key that is no longer needed. This process includes all materials (both physical and electronic documents) associated with a key. This should be done in a secure and proper manner so that after the key is disposed, no other remaining information can be used to recover the disposed key.

Note that while all of these processes require attention, the majority of GKMFs for group communication (Mittra, 1997), (Wong et al., 1998), (Waldvogel et al., 1999), (Hardjono et al., 2000a), (Baugher et al., 2003), (Hardjono and Weis, 2004), (Baugher et al., 2005) and key management standards ISO/IEC 8732 (ISO, 1988), ISO/IEC 11770-1 (ISO, 1996a) and ISO/IEC 11770-2 (ISO, 1996b) are mainly concerned with distribution and updating (re-keying) of cryptographic keys. Other aspects of key management are implicitly assumed to be available and securely managed by trusted entity(s), since their provision is handled by generic key management processes that are not specific to group key environments.

4.7 Important Features

We have identified the main entities and processes that form a GKMF. In this section, we identify features that in our opinion a good GKMF should have.

4.7.1 Historic Development

Note that the initial designs of GKMFs, such those in (Ballardie, 1996), (Caronni et al., 1996), (Harney and Muckenhirn, 1997), (Harkins and Carrel, 1998) and (Maughan et al., 1998) considered only the basic features sufficient to support multicast group communication, without considering further needs for security (note that IP multicast was never intended to provide secure multicasting).

From the early 1990s we have witnessed the emergence of applications (see *Section 2.3*) that require secure multicast technology. Thus, the need for secure, reliable and scalable GKMFs has become increasingly important.

Another aspect of this is that, as the demands for group-based applications have increased, service providers have realized that they want more out of the multicast facility. In particular there has been demand to:

(a) control access to information to certain groups of hosts.

(b) restrict access to valid group members during specific time periods (while they are registered (valid) members of a multicast group).

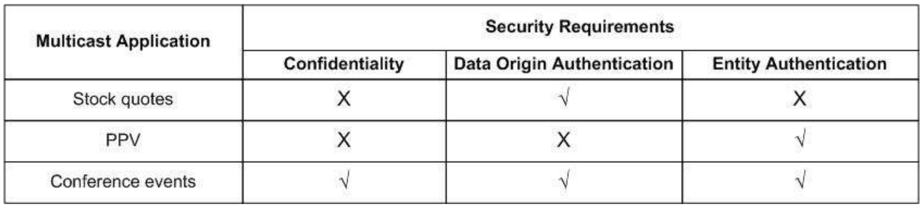
(c) preclude members who have ceased to be group members of a multicast group.

4.7.2 Different Features for Different Applications

An additional challenge for GKMF design is that different applications typically require different features. There is normally no such thing as one solution fit for all applications, and GKMFs for secure multicast group solutions are no exception. A particular design of a GKMF may be sufficient to meet certain requirements of some multicast group applications, while lacking for others.

Consider three applications mentioned in *Section 2.3*: *stock quotes distribution, PPV (Pay per View) channels* and *conference events.*

- *Stock quotes.* A service providing stock quote information may not require confidentiality (since such data is often public), but the recipients may wish to ensure the integrity of data received and that it originates from a valid sender.



Multicast Application	Security Requirements		
	Confidentiality	Data Origin Authentication	Entity Authentication
Stock quotes	X	√	X
PPV	X	X	√
Conference events	√	√	√

Table 4.1: Multicast applications and their security requirements

- *Pay per view.* Viewers of PPV channel (the recipients) may not care about source authentication (as long as they obtain the right channels). However, the PPV service providers may wish to restrict access to channels only to users who are actually paying for the service. Note that, due to the nature of application, both viewers and service providers may not require confidentiality services.

- *Conference events.* Conference events, where only members who have registered for a conference should be granted access to conference materials, may require a confidentiality service. Conference organizers may wish to control access to materials and ensure that only registered members are able to view them.

We summarize the security requirements of the aforementioned applications in *Table 4.1*.

The provision of these different security requirements all involve some kind of cryptographic keys being managed properly and securely within a GKMF, although the precise security services required depend on the requirements of particular multicast applications.

4.7.3 Specification of Features

We attempt here to identify the key features that a GKMF should have. From various studies on existing GKMFs (Mittra, 1997), (Harney and Muckenhirn, 1997), (Wong et al., 1998), (Waldvogel et al., 1999), (Wallner et al., 1999), (Hardjono et al., 2000a), (Hardjono et al., 2000b), (Decleene et al., 2001), (Baugher et al., 2003), (Hardjono and Weis, 2004) and (Baugher et al., 2005), we have extracted the important features of a GKMF and divided them into two parts; essential properties that every GKMF should have and desirable properties that are optional depending on the specific requirements.

4.7.3.1 Essential features

As discussed in *Section 4.7.2*, different applications may require different properties depending on their specific requirements. We thus classify the essential features that a secure GKMF should have into two further categories:

(1) *Independent of application*

Independent of application in place, the essential features that a secure GKMF should have are:

- *Dynamic group membership policy.* To allow a framework to be as flexible as possible group members should be free to join and/or leave any time during any session throughout the multicast group lifetime.
- *Backward and forward Secrecy.* A good GKMF should ideally provide these on a default basis, maintaining the secrecy of information to valid group members at all times.
- *Dynamic and efficient re-keying processes.* To efficiently manage any re-keying that will need to occur. Re-keying processes should be conducted without disrupting any on-going communications.
- *Scalability.* Group communications can potentially involve tens of thousands of members, many of whom may be constantly joining and leaving groups. A GKMF should thus be scalable with respect to the efficient distribution and management of keying material.
- *A trust model.* A trust model is crucial for a GKMF in order for it to properly function. This should include a security architecture that is compatible with the existing network protocols, as well as scalable and transparent to higher level applications and services. This also includes determination of delivery point(s) of keys.

- *Reliable and trustworthy key manager.* Whether key managers are centralized or distributed, they should be sufficiently reliable that other entities (including group members) trust them.

(2) *Dependent of application*
Dependent of application, the essential features that a secure GKMF should have are:

- *Secure data exchange.* This includes mechanisms for protecting transmitted data, regulating group members' access to data and verifying to any member the nature of the group session in which they participated.
- *Group and member authentication.* Apart from the initial registration with a trusted group manager upon joining a multicast group, GKMF protocols ought to consider not just verifying that a member is valid and belongs to a valid group (group authentication) but may also choose to verify the actual member (member authentication) participating in the group communication.

4.7.3.2 Desirable features

In this section, independent of application, we identify other desirable but optional features of a GKMF as follows:

- *Minimizing computational and storage efforts.* Keeping to a minimum the amount of computation that needs to be done, and keys that need to be managed and securely stored, by all communicating entities during a multicast group communication.
- *Minimizing traffic implosion.* Keeping to a minimum the number of messages that needs to be exchanged during GKMF protocols.
- *Reduced trust in third party and intermediate nodes.* To minimize the reliance upon third party nodes which may be needed to support a secure GKMF.

- *Minimizing risk of attack vulnerabilities.* Protecting and minimizing the risk of group data and any keying materials from both passive and active attacks that would compromise the security objectives of the group communication.
- *Minimizing risk of colluding members.* Minimizing the impact of group members who exchange information in order to gain additional unauthorized access to the group data traffic.
- *Coping with system and network failures.* Any good and reliable communication architecture should be able to deal with system and network failures, either caused by human errors or natural disasters.

4.8 Summary

In this chapter, we described the basic components of a GKMF and established desirable features which are generic for any networking environments (wired or wireless networks).

In the next chapter we provide additional challenges for deploying multicast group communications in wireless mobile networks.

Chapter 5

GKMF: Design Challenges in Wireless Mobile Environments (WMobEs)

This chapter briefly looks at the additional challenges faced by establishing GKMFs for wireless mobile environments (WMobEs).

In *Section 5.1* we identify several types of WMobEs. In *Section 5.2* we look at inherent characteristics of WMobEs which introduce new challenge for multicast group communication in such environments. In *Section 5.3* we identify specific problems associated with the establishment of multicast group communication in WMobEs.

5.1 Types of Wireless Mobile Environments

While issues concerning group communication are widely researched and the development of secure multicast becoming more pertinent in wired networks, its implementation in mobile environments (wireless networks) is still in its infancy. Applications and services which are available in wired networks should also be made available in wireless networks and vice versa. There are similar expectations for providing secure and reliable communication in both environments.

Main types of WMobE can be categorized as:

(a) *Fixed-based networks*
A collection or a group of wireless mobile nodes communicating or using the services usually provided by corporate enterprise networks or small office home offices (SOHO) over wireless mediums such as wireless local area networks (WLANs), or cellular-based networks such as GSM or UMTS (Lin and Chlamtac, 2001), (Hillebrand, 2002). This kind of group communications operates with the help of fixed infrastructures such as base stations, access points or satellites.

(b) *Non Fixed-based networks*
Operating without the help of any infrastructure, non fixed-based networks can be further classified into:

- *Adhoc networks.* A collection of wireless mobile nodes communicating among themselves, possibly over multi-hop paths, without the help of any infrastructure such as base stations or access points (IIyas, 2003), (Michiardi and Molva, 2006).
- *Sensor networks.* A special form of adhoc networks, sensor networks consist of a collection of individual nodes (usually battery operated), each of which transmits data signals also without the help of any infrastructure. Typical use of an individual sensor is for collecting specific data such as sudden changes in climate across a geographical area (Wadaa et al., 2004).

We note that the WMobEs concerned with our work rely on fixed network infrastructures where reliable entities such as domain key managers and area key managers are assumed available. We will not consider ad-hoc or sensor networks in this thesis.

5.2 Inherent Characteristics of WMobEs

Our list of desirable features for a GKMF in *Section 4.7.3* is generic, and equally applicable to wired and wireless networks (WMobEs). However, there are three main aspects in which WMobEs are significantly different. We look at each of these in the following subsections.

5.2.1 Non-fixed and Wireless Network Connectivity

In comparison to fixed wired networks, wireless mobile networks have some very different security issues:

- *More susceptible to security attacks.* Compared to wired data transmissions, wireless data transmission is more susceptible to attacks discussed in *Section 4.4*, particularly passive attacks such as eavesdropping and monitoring of data traffic. These attacks are more acute in WMobEs since they are easier to carry out in such environments because an adversary does not have to physically tap into a network (Vines, 2002) and (Nichols and Lekkas, 2002). This could result in further security breaches if certain measures are not in place. For example, messages or group data that are not encrypted can easily be read and understood by an adversary who is eavesdropping on the group data traffic.

- *Trust within foreign networks.* As mobile environments allow members to move around, changes in location may require group members to occasionally communicate via foreign networks that cannot always be trusted. This affects the amount of trust to impart on governing entity(s) within foreign networks (including how much information they want to share).

  Likewise, foreign networks may want to consider the amount of trust they want to place on the visiting mobile hosts, since they may gather information about the local security services for the areas they visit, which can lead to security threats if not contained.

- *Susceptible for disconnections.* Compared to wired networks, transmissions in wireless networks have high tendencies for disconnection. Host mobility in such environments aggravates this issue further. Frequent disconnections during data transmission may cause unnecessary disruptions to group communication. In this case, minimal delay during key management processes may be useful in such environments.

5.2.2 Nature of Mobile Devices

Many mobile devices such as mini PDAs or smart phones exhibit special characteristics with regards to having limited power consumption (or energy supply), limited data (key) storage as well as having lower processing speed, which hinder intensive and heavyweight processing. With such constraints, the design of a GKMF should employ:

- As little cryptographic computation as possible.
- Minimal messaging bandwidth usage (in other words, minimum message exchange between communicating entities).
- Efficient storage (normally associated with minimizing numbers of keys that a mobile device needs to store).

5.2.3 User (Host) Mobility

Wireless mobile environments allow user (host) mobility. As group members are allowed to move between areas, the mobility issue exhibits problems that do not occur in wired networks such as:

- *Hand-off operations.* When group members move from one area to another, some kind of hand-off operation from the current area to the visited area (the area where the member is moving to) is required.
- *Management of keying material.* Problems pertaining to management of cryptographic keys needed during host mobility include deciding who governs the moves and who keeps track of keying material.

- *Network disconnection.* As discussed in *Section 5.2.1*, network transmissions between communicating entities may change over time, and be prone to failure (disconnection).

The issues discussed so far are, to an extent, the common security problems associated with wireless mobile networks. The implementation of multicast group communications in such environments introduces further issues that need to be addressed by a GKMF. We look at these issues in the next section.

5.3 Issues Specific to Group Communications in WMobEs

We have mentioned the main security issues pertaining to group communications in *Section 3.2*. In this section we present specific security issues concerning group communications for deployment in wireless mobile networks.

5.3.1 New Reasons for Joins and Leaves

Group members no longer just join and leave a multicast group for reasons discussed in *Section 4.3* but are also allowed to join and leave as they move between areas while still remaining in a group session. Thus, in addition to the processes discussed in *Section 4.3.2*, a specific protocol is required to govern members' movement between areas, which is also referred to as *host mobility*.

The process of a member moving to other areas may be treated as a *leave* from one area followed by a *join* to another. Any member who wishes to move will need to notify a key manager, prior to moving. Similarly to a member joining and a member leaving (see *Section 4.3.2*), a member moving to another area may require the provision of backward and/or forward secrecy (since different areas may have their own security requirements). In this case, controlling access to visited areas may be necessary.

In dynamic environments, group members may be allowed to freely move between areas while still remaining in a group session.

5.3.2 Additional Key Management to Support Mobility

The generation of new keying material may be required in order to support host mobility. For example:

(a) Moving members may still hold cryptographic keys of the areas they visited even after they leave a multicast group, which may lead to compromise.

(b) Host mobility may require group members to occasionally communicate via a foreign network (the visiting area) that may not be fully trusted. Thus, it is important to ensure that group members that are moving from one area to another are protected (via different sets of keys).

(c) Group members that move between areas may gather the area's local security information. It is imperative to ensure that the area is protected from members who are moving from one area to another in order to collect the security information (keys) of each area for malicious purposes.

5.3.3 Performance Requirements

We have looked at the security and key management requirements in *Section 4.7.3*, which also apply for designing a GKMF for WMobEs. In this section, we discuss the performance requirements of a GKMF in a WMobE.

The performance requirements can be divided into several categories as follows:

- **Computation, energy and storage requirements**. All computations related to group security should be resource efficient, since mobile nodes are usually equipped with limited power and storage, which restricts their ability to do rigorous computations and to store a large amount of data. Therefore, any security-related operations to be done by mobile devices are to be kept as low as possible. This also includes the number of keys that each member needs to store throughout a multicast group session.

- **Communication and bandwidth requirements**. In WMobEs, since frequent connection cut-offs are likely to happen and the bandwidth available is limited compared to wired environments, the number of messages exchanged (such as between key manager entities sending and/or receiving keys and other security related information) should be kept as low as possible. As battery life is also a big problem in many mobile devices, less messages that need to be processed and sent may be necessary since more messages will drain batteries life.
- **Mobility requirement**. All security-related information associated with each member during his mobility, including the distribution of new keys or key updates, needs to be managed in a proper manner so that the performance (quality of service) of the framework should not be significantly reduced by host mobility.

5.4 Summary

We have looked at the additional challenges faced by establishing GKMFs for wireless mobile environments.

In the next chapter we provide a generic GKMF model suitable for such environments.

Chapter 6

A Generic GKMF Model for Wireless Mobile Environments

This chapter proposes a generic GKMF model for WMobEs. The purpose of this generic model is to act as a blueprint within which we can later specify particular frameworks.

The generic framework is divided into two main parts. *Section 6.1* describes the *main components* of the framework. *Section 6.2* presents the *main protocols* (or processes) of a GKMF that are needed for multicast group operations in WMobEs. Part of the work in this chapter has been published in (MatKiah and Martin, 2006).

6.1 Main Components

In this section we describe the main functional components that a GKMF should have for secure multicast group communication. This includes the general architecture of typical entities that form a GKMF, their relationships and cryptographic keys used. It also includes other aspects of the framework such as options for assigning key managers, and group membership policy.

6.1.1 General Architecture

The main components that comprise an architecture can be listed as follows:

(a) **Main entities**

The main entities identified within a framework for group key management are typically:

- *Server(s).* Typical roles of servers are as key managers, group managers, group controllers, or as supporting nodes for multicast group communication.
- *Hosts.* Hosts or group members are the lowest level entities within a GKMF. Also referred to as *key users*, group members engage in the actual group communication. Group members consist of at least one *sender* of data, and one (or more) *recipients* of data.

Figure 6.1 illustrates a basic model for group key management with a central entity. From the figure, a key server (which can also be referred to as group manager) is the central managing entity for group members; one as a sender and another as a recipient. Both members need to establish security parameters prior to group communication, and this is done through key and security association (SA) management (denoted by ⟷ arrows). Any subsequent communications such as key updates that need to occur are done through a control channel between the key server and the group members (denoted by *dashed* arrows). The actual group communication among the group members is denoted by a **bold** ⟶ arrow.

(b) **Domain(s) and Area(s)**

Domain(s) and area(s) in wireless mobile environments (WMobEs) can be described, as follows:

- *Domains.* Logically, a domain can be viewed as a bigger system which comprises a group of subsystems, which can consist of hundreds or thousands of services. From another perspective, a domain can cover a physical large geographic area (such as a UK region), which consists

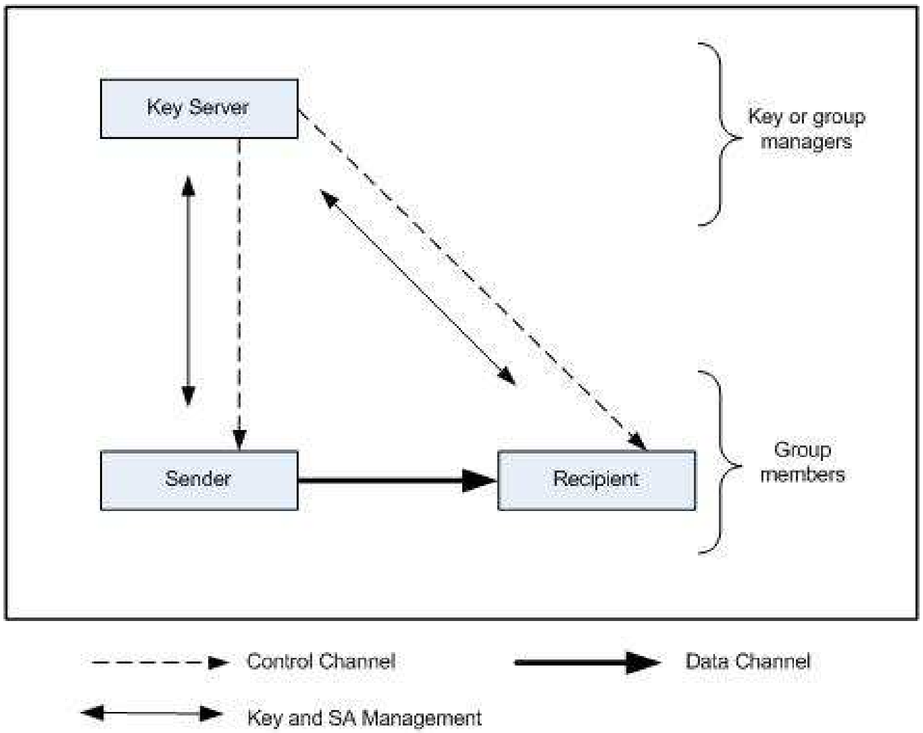


Figure 6.1: A basic model for group key management with central entity.

of smaller areas (such as counties). Each domain may have its own purpose, common goals and objectives.

- *Areas.* Smaller versions of domains, areas can be viewed logically as subsystems which operate under the governance of bigger systems (such as *domain* system). Similarly, an area can cover a physical geographic area (such as one or more counties) which can be part of larger system (such as a region). While all areas in a domain may share the domain's general goals and objectives, each area may have its own unique requirements.

  Typically for the purpose of key management, in a wireless mobile network (such as GSM network), a domain can be one large physical area which may be governed by physical entity (such as a group controller or a key manager). For efficient key management, the domain can be further divided into smaller manageable physical areas, each of which may have its own governing entity.

WMobEs allow group members to move from one place to another while still participating in a group session. For ease of managing host mobility in such environments, group members are typically placed in these manageable areas.

As WMobEs can consist of multiple domains and areas that can be overlapping logically and physically, we use *intra-domain* to describe relationships within a domain, and *inter-domain* to describe relationships between two (or more) distinctive domains.

An example of domains and areas is depicted in *Figure 6.2*. Domain D_A and D_B each consists of several areas. The *inter* and *intra* domain relationships are shown between and within the domains.

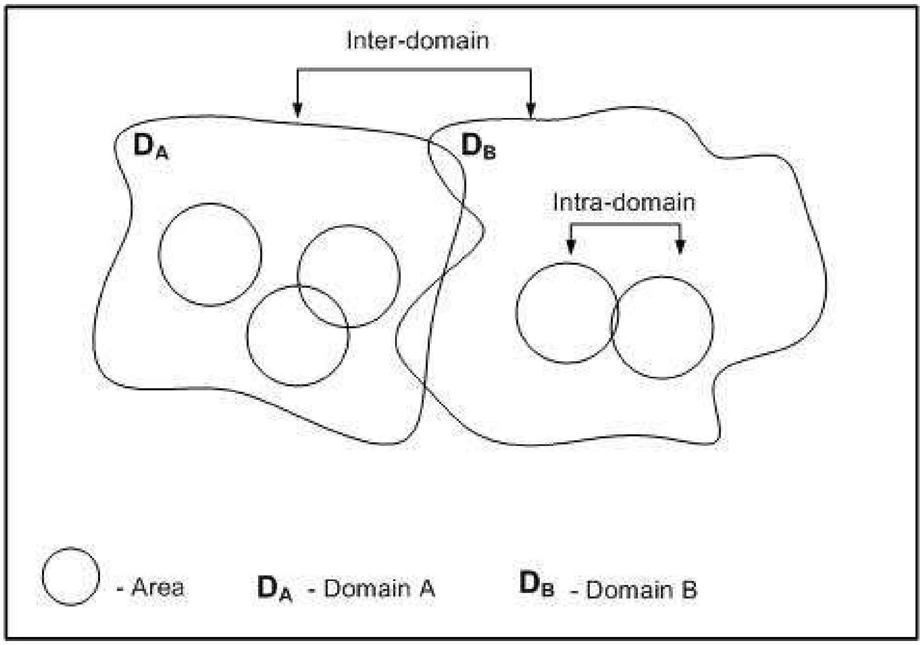


Figure 6.2: An example of domains and areas.

(c) **Placement of entities**

The entities in the framework identified in (a) must be placed in the domains and areas identified in (b). As discussed in (b), a domain can be further divided into smaller (or multiple) areas (see *Figure 6.2*). Thus, these areas need to be placed within a domain and group members within these areas. Domain(s) and areas may have their own managing entity.

Typical placement of entities based on a centralized framework is illustrated in *Figure 6.3*. Here, domain j is depicted to consist of at least two independent areas, which are *Area a* and *Area b*, with one managing entity at the domain level referred to as the *domain entity*. Similarly, at the area level, each area consists of one managing entity referred to as an *area entity*. Group members M can be positioned in any of the areas in domain j, and each area entity is responsible for group members residing in its area.

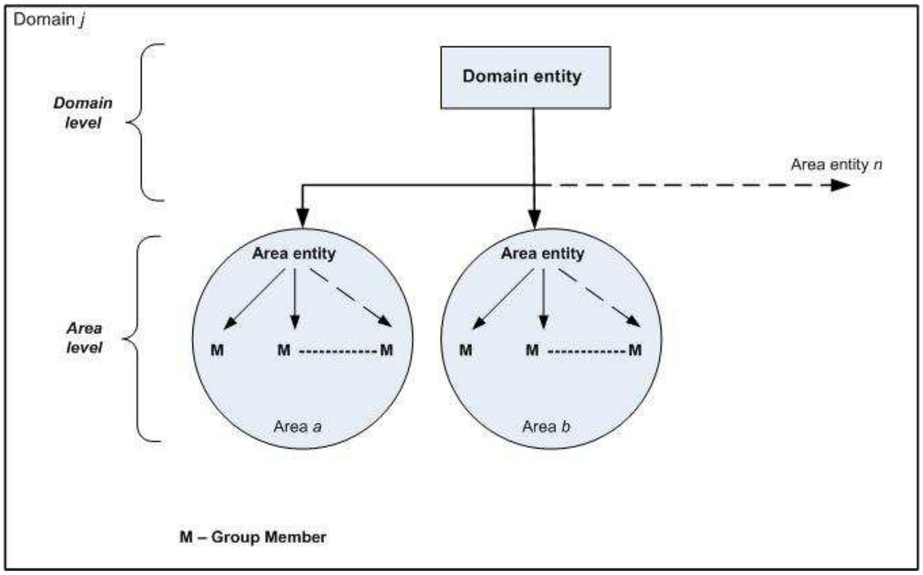


Figure 6.3: An example of placement of entities in a domain j.

(d) **Trust Relationships**

Trust relationships between the entities need to be specified. Typically, trust relationships within a GKMF revolve around key managers (as the main key distributors). Key managers are often configured and maintained by human administration and implemented using secure technology, since they represent the best point of attack.

As mentioned in *Section 4.2*, all entities involved in multicast group communication in a centralized scheme trust the centralized key manager in the domain, which is the primary source of security parameters (such as cryptographic keys) needed for group communication. When the need arises, this entity can also take on the role of a certification authority.

On the other hand, distributed schemes require having multiple entities acting as key managers, who are jointly trusted for the generation and distribution of the security parameters needed for the group communication.

(e) **Types of Key**

The types of key used in the framework need to be specified. As mentioned in *Section 4.3.2*, cryptographic keys can be symmetric, asymmetric or a combination of both.

Where symmetric keys are used, they can typically be classified into:

- *Traffic keys.* Primarily used for securing actual data communications (also referred to as *traffic encryption keys* or *traffic protection keys*), and shared amongst all group members of a particular multicast group. Typically, a traffic key is unique for a multicast group.
- *Domain keys.* These keys are typically shared and used by key managers for secure distribution of *traffic keys* within a domain (also referred to as *group keys* or *key encryption keys*). In WMobEs where multiple domains may exist, every domain should have its own domain key.
- *Area keys.* These keys are typically unique to an area and shared between key managers and group members residing in the area (also referred to as *area control keys*). Used for secure distribution of keys by an area key manager to members within an area. They also form auxiliary keys (at the area level) which may be used by members to send secure messages amongst one another within that particular area.
- *Individual keys.* These keys are long-term secrets between key managers and group members. These keys typically need to be established prior to any request to create multicast groups.

In cases where asymmetric keys are used, all entities are normally assigned asymmetric key pairs. Each member is assumed to have a copy of the public-keys of relevant key managers. Key managers are also assumed to have copies of public keys of any other key managers they need to communicate with.

6.1.2 Group Membership Policy

As mentioned in *Section 3.2.1*, group membership can be categorized as:

(a) *Static, or closed membership;*

(b) *Dynamic, or open membership.*

As noted in *Section 4.3.2*, we normally assume that dynamic membership is supported in any GKMF for WMobEs.

6.1.3 Design Approach

As mentioned in *Section 4.2*, this can be:

(a) *Centralized scheme,*

(b) *Distributed scheme, or*

(c) *Hybrid scheme.*

6.2 Main Protocols

In this section we describe the main protocols needed for designing a GKMF for deployment in a WMobE. While many of these are generic GKMF protocols mentioned in *Section 4.3.2*, we also need a protocol for *host mobility* to govern the movement of group members between areas throughout their membership of a multicast group.

6.2.1 Protocol for Creating New Group

This protocol governs the creation of new multicast groups. It also includes the initial registration of group members to a multicast group and initial distribution of keys to that particular member. Note that, typically, any host who requests the creation of a multicast group is considered to be the first host to register (or join) the group.

At this point, the information related to group membership policy, as well as the relevant keys (see *Section 6.1*) necessary for group communication, are determined.

6.2.2 Protocol for New Member Joining

As mentioned in *Section 4.3.2*, this process is similar to the initial registration of group members. In particular, this protocol governs new joins of group members into a multicast group. It also includes the distribution of all cryptographic keys needed for the group communication to the newly joined members.

As mentioned in *Section 3.2.2*, different membership policy affects the way we manage the cryptographic keys. Thus, this protocol introduces two options which can be adopted in admitting any hosts to become group members of a multicast group:

(a) *Members joining with backward secrecy*
In the first option, typically when a newly joined member is to be prevented from accessing previous group traffic or old group keys, all cryptographic keys associated with the multicast group, including the area where the group member is residing, need to be re-keyed.

(b) *Members joining without backward secrecy*
In the second option, no such restriction applies, and re-keying may not need to occur. New group members are given the same set of keys that are currently used by existing members.

6.2.3 Protocol for Existing Member Leaving

This protocol governs existing members leaving a multicast group. As mentioned in *Section 4.3.2*, members leaving can be:

(a) *Voluntary, or*

(b) *Non-voluntary.*

Like members joining, two options can be adopted due to members leaving:

(a) *Members leaving with forward secrecy*
In the first option, if controlling access to future group communication is necessary then re-keying needs to occur whenever an existing member leaves a multicast group. All relevant keys will need to be generated and distributed to all remaining group members.

(b) *Members leaving without forward secrecy*
In the second option, no such restriction applies. Re-keying may not need to occur.

6.2.4 Protocol for Member Moving to Other Areas

As mentioned in *Section 5.3*, we need a protocol for *host mobility*. This protocol governs members moving to other areas. Hand-off operations between affected areas (areas where group members are moving from and moving to) may need to occur prior to members moving in order to establish relevant keys.

As the security requirements for a member moving to another area is quite similar to that of a new member joining (see *Section 5.3*), two options can be adopted due to host mobility:

(a) *Members moving with backward secrecy*
In the first option, typically if a moving member is to be denied access to

an area's security information (the area where the member is moving to), all cryptographic keys associated with that area need to be re-keyed.

(b) *Members moving without backward secrecy*
In the second option, no such restriction applies, and re-keying within an area may not need to occur. Moving members are given the same set of keys that are currently used in that area.

6.2.5 Protocol for Re-keying

As mentioned in *Section 4.3.2*, re-keying may need to occur due to:

(a) *Group membership change.*

(b) *Periodic re-keying.*

(c) *Expiration of cryptographic keys.*

(d) *Compromised keys.*

WMobEs may also require re-keying due to *host mobility* (see *Section 5.3*). Each of these processes results in all group members obtaining the new cryptographic keys needed for secure group communication.

In WMobEs where domains and areas may have their own managing entities, re-keying can be divided into two levels:

(a) *At the domain level*
Re-keying at the domain level is normally initiated and controlled by a domain manager. For example, for the provision of backward secrecy, whenever a new host joins a multicast group, the group's traffic key must be re-keyed. The domain governing entity will have to initiate re-keying of the traffic key by generating a new traffic key and delivering it to all area manager entities and group members.

(b) *At the area level*

Re-keying at the area level is initiated and controlled by an area manager governing that area. Similarly, for the provision of backward secrecy, the area key of an area (where the new join occurs) must be re-keyed. The area manager will have to initiate the re-keying of its area key by generating a new area key and sending it to group members in that area (including the newly joined member).

6.3 Summary

We have proposed a generic GKMF model for group communication for WMobEs. In the next chapter, we will illustrate how a number of GKMFs in the literature fit into this generic model.

Chapter 7

Existing GKMFs

This chapter looks at existing GKMFs. We show how they fit into our generic model of Chapter 6*, and identify where they are deficient for our purpose of establishing a GKMF for a WMobE. None of them provide the full functionality we require, but this exercise will influence our design of a specific GKMF in the remaining chapters of the thesis.*

In *Section 7.2* we present the existing frameworks. In *Section 7.3* we show how these frameworks fit into our generic model.

7.1 Introduction

In this chapter, we look at existing architectures that are relevant in identifying certain attributes that are useful for our purpose. A number of GKMFs have been proposed in the literature, including (Mittra, 1997), (Harney and Muckenhirn, 1997), (Wong et al., 1998), (Waldvogel et al., 1999), (Wallner et al., 1999), (Hardjono et al., 2000a), (Hardjono et al., 2000b), (Baugher et al., 2003), (Hardjono and Weis, 2004) and (Baugher et al., 2005).

We will present a brief review of the architectures of (Harney and Muckenhirn, 1997), (Mittra, 1997), (Wong et al., 1998), (Hardjono et al., 2000a), (Hardjono et al., 2000b) and (Baugher et al., 2003), since they particularly influenced our later design.

We choose these proposals instead of others (as mentioned) because of the following reasons:

- GKMP-A (Harney and Muckenhirn, 1997) was the first GKMF proposed for multicast communication. As such, it provides a benchmark against which subsequent research can be measured.
- Key-graph (Wong et al., 1998) was the first framework to propose the notion of secure group communication based on a hierarchy of keys. Its idea of having high level keys to protect low level keys is interesting from the perspective of designing scalable frameworks.
- Iolus (Mittra, 1997) was the first GKMF to introduce the idea of subgroups to mitigate scalability problems, due to re-keying events that may need to occur during group operations.
- F-MSEC (Hardjono et al., 2000a) proposed the idea of regions, where each region associates with a different cryptographic key. The idea of multiple regions is natural for most wireless mobile networks and so F-MSEC is relevant to our study.
- Intra-domain GKMP (Hardjono et al., 2000b) is based on the F-MSEC (Hardjono et al., 2000a) framework, and extends the idea of regions by introducing domains and areas, each of which is independently managed. This concept fits well with the types of wireless mobile architecture that we have in mind.
- GKM-A (Baugher et al., 2003) is an extension of GKMP-A, and introduced the idea of exchanging multiple security associations (SAs) to establish secure group communication. It also includes improvements in processes such as registration of group members that are of relevance to our requirements.

We do not consider other frameworks which exhibit similar features to the six chosen frameworks. For instance, other tree-based frameworks such as (Waldvogel et al., 1999) and (Wallner et al., 1999) are similar to Key-graph (Wong et al., 1998). We observe that works in (Waldvogel et al., 1999) and (Wallner et al., 1999) do not alter the original idea of Key-graph, but rather improve other aspects of group communication which are covered by our chosen frameworks. The frameworks in (Hardjono and Weis, 2004) and (Baugher et al., 2005) are closely related to GKM-A, Intra-domain and GKMP-A.

Note that the majority of the chosen proposals are part of IETF WG (MSEC, 2007) research.

7.2 Related Frameworks

In this section we present the existing frameworks relevant to our purpose:

7.2.1 GKMP-A

GKMP-A (Harney and Muckenhirn, 1997) was the first proposal to introduce protocols for group key management in multicast communication.

The framework can operate in two different modes; *sender initiated*, or *receiver initiated*. In *sender initiated* mode, the sender will act as the group key controller of the multicast group. Otherwise, it will be one of the receivers of the multicast group who controls the key.

The sender-initiated model operates in the form of a group key management application, operating on behalf of the originator as well as the group key controller of a multicast group. In the receiver-initiated model, a group member is made responsible for initial group key establishment, as well as the periodic generation and dissemination of new keys.

7.2.1.1 Main Architecture

GKMP-A does not require a centralized key manager (with the exception made for an entity who manages the distribution of security certificates).

The main entities involved in the protocol are:

(a) *Group key controller (GKC)*
The GKC is appointed by the group key management application to be the main group key controller of the group. Either the sender or one of the receivers will be the GKC, depending on the mode of operation. The GKC manages the aspects of key management, including the generation and distribution of cryptographic keys to all members of the group.

(b) *Security manager*
Responsible for creating and distributing authentic identification and security information (such as certificates needed for communicating entities).

We observe that because of the introduction of two operational modes, which then determine who will be the GKC for the multicast group, GKMP-A avoids having a centralized key manager as the only parties involved in key management are the same parties (group members) who will be participating in the group communication.

GKMP-A does not address other aspects of a framework such as placement of entities in the architecture, group membership policy, and trust relationships.

7.2.1.2 Keys

Keys that are used in the protocol are classified into three categories:

(a) *Group key packet (GKP)*, which contains two keys:

- *Group traffic encrypting key (GTEK)* is the current traffic key for encrypting group data traffic.
- *Group key encrypting key (GKEK)* is the future key used for secure distribution of a new GTEK.

(b) *Session key package (SKP)*, which is unique for each member and contains two keys:

- *Session traffic encrypting key (STEK)* is the key used for encrypting unicast data traffic between a member and the key controller.
- *Session key encrypting key (SKEK)* is the key used for secure distribution of a new STEK to a particular member.

(c) *Group rekey package (GRP)*
The GRP contains a new GKP encrypted with a key encrypting key (GTEK or SKEK) and signed by the originator's private signature key.

A GKP is created by the application prior to the establishment of an SKP and GRP.

Although GKMP-A claims to use a combination of both symmetric and asymmetric cryptography, we note that the main focus is on symmetric cryptography, and asymmetric cryptography is only used to provide the originator's (GKC) certificate. It is assumed that all group members have a means to verify the originator's signature.

7.2.1.3 Main Processes

Based on the two aforementioned operational modes, two processes are introduced:

(a) *Generation and distribution of keys*
For *sender initiated* operation, group keys (see *Section 7.2.1.2*) are generated by the application in the form of a GKP. The GKP is distributed to

one of the group members selected by the application, which then becomes the GKC of the multicast group. The GKC then contacts each member of the group, creates an SKP (which is unique to each member), creates a GRP (which contains the GKP) and sends these to each of the group members.

For *receiver initiated*, a GKP is created by the application with the first member to initiate contact, which is then distributed to the member. The rest of the operation is similar to *sender initiated*, where other group members are each keyed with an SKP and a GRP.

(b) *Re-keying of group keys*
Re-keying of group keys was briefly addressed. For both modes of operation, when a re-key is required, the application selects a member, creates a new GKP and a new GRP, and distributes the keys to other members.

With the exception of these processes, GKMP-A does not address other GKMF protocols such as creation of multicast group, or member joining and leaving protocols.

7.2.2 Key-graph

Key-graph (Wong et al., 1998) was the first proposal to conduct secure group key management by employing a hierarchy of keys in order to reduce the number of messages required for re-keying.

The notion of a secure group is organized in the form of a logical tree, as illustrated in *Figure 7.1*.

From *Figure 7.1*, there are two main groups of *nodes*; K-nodes and U-nodes. Every K-node in the tree is assigned a key, and each user ($U1..U5$) is associated with a U-node and placed as a leaf node. Each user is assigned keys along the path from its leaf to the root node. For example, user $U1$ is assigned the keys $K1$, $K123$, $K12345$.

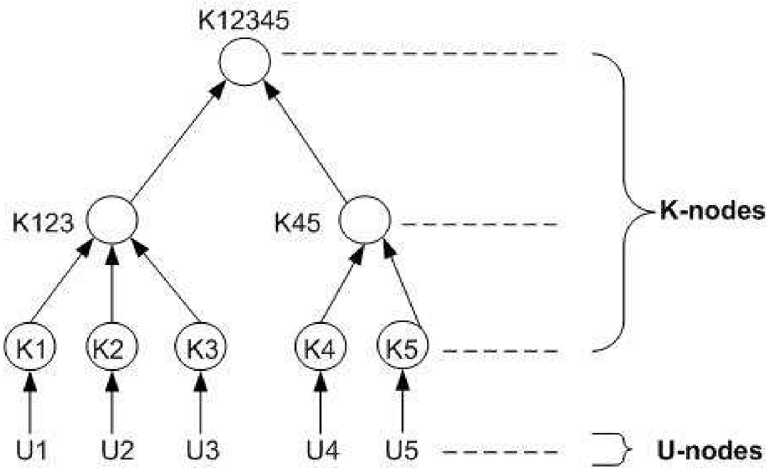


Figure 7.1: An example of Key Tree Graph.

7.2.2.1 Main architecture

The main entities involved in this architecture are:

- *User*: Consisting all group members.
- *A server (s)*: With primary responsibilities for managing group members and keys, it is assumed that the server also acts as the group controller for all group users.

Key-graph does not address other aspects of a GKMF such as placement of these entities in the architecture, trust relationships, and group membership policies.

We observe that Key-graph adopts a centralized scheme in its design.

7.2.2.2 Keys

Symmetric keys that are used in this architecture are:

(a) *Individual keys*

Also referred to as *leaf keys*, an individual key is shared only between a user and the key server (or the group controller). Used primarily for communication between a user and the key server. For example, $K1..K5$ are individual keys, each associated with user $U1..U5$ respectively (see *Figure 7.1*).

(b) *Subgroup keys*
Also referred to as *internal keys* (keys which are in-between leaf and root nodes) or *key encrypting keys* (since they are normally used for encrypting other keys). For example, $K123$ and $K45$ are subgroup keys, associated with subgroups $\{U1, U2, U3\}$ and $\{U4, U5\}$ respectively (see *Figure 7.1*).

(c) *Group keys*
Also referred to as *root keys* (the highest nodes in the tree hierarchy), a group key is shared between all users and the key server. A group key is primarily used for encrypting group data and is also called a *traffic encryption key*. For example, $K12345$ is a group key shared by all users (see *Figure 7.1*).

These symmetric keys are generated and distributed to users prior to group communication.

7.2.2.3 Main Processes

The main processes described in the framework are re-keying due to hosts joining and leaving. When a new user joins a group, the K-nodes along the path from the new leaf (which is added to assign the new user) to the root are re-keyed with new keys.

Based on *Figure 7.2*, when $U6$ joins the subgroup $\{U4, U5\}$, subgroup key $K45$ and group key $K12345$ need to be re-keyed as new $K456$ and $K123456$ keys. This results in all users of the subgroup (where the new join occurs) including the new user $U6$, obtaining new $K456$ and $K123456$ keys, and all other subgroup users $\{U1, U2, U3\}$, obtaining the new $K123456$ key.

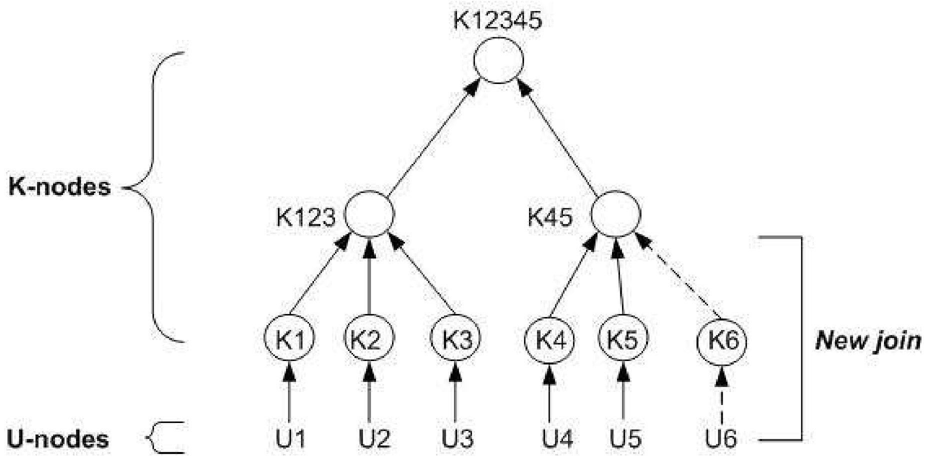


Figure 7.2: Key Tree Graph: A new join by user $U6$ into the group.

When a user leaves, a similar re-keying process applies. For example, when user $U6$ leaves its subgroup $\{U4, U5, U6\}$ (see *Figure 7.2*), all users in its subgroup (where the leave occurs) obtain a new subgroup key $K45$, and all users excluding the leaving user $U6$, obtain a new group key $K12345$.

With the intention of reducing the number of messages sent during re-keying, three approaches for re-keying are suggested:

(i) *User-oriented*

In this approach, for each user, the key server sends a re-key message that contains the new key, encrypted with a key held by the user. For example, during a new join of user $U6$ (see *Figure 7.2*), the server needs to send users:

- $U1, U2, U3$, new group key $K123456$ encrypted with $K12345$.
- $U4, U5$, new group key $K123456$ and new subgroup key $K456$ encrypted with $K45$.
- $U6$ (new user), new group key $K123456$ and new subgroup key $K456$ encrypted with $K6$.

(ii) *Key-oriented*

In this approach, each new key is encrypted individually (except keys for the joining user). A user may have to get multiple rekey messages in order to get all the new keys it needs. For example, as in the *user-oriented* case, when a user $U6$ joins the group, the server sends a new group key $K123456$ and new subgroup key $K456$ to users $U4$ and $U5$ separately, as follows:

- New group key $K123456$, encrypted with $K45$.
- New subgroup key $K456$, encrypted with $K45$.

It is observed that in order to save re-key costs, these re-key messages can be combined and sent to users as one message. Note that these keys need to be sent to the new user $U6$ separately.

(iii) *Group-oriented*
In this approach, each new key is sent to users (except keys for the joining user) as a *group*. In this case the server sends a single rekey message containing all new keys via multicast to the entire group. For example, as in the *user* and *key-oriented* cases, when a new user $U6$ joins, the server needs to send users:

- $U1..U5$, new group key $K123456$ encrypted with $K12345$.
- $U4$ and $U5$, new subgroup key $K456$, encrypted with $K45$.

Note that these keys are sent in one combined multicast message. As with the *key-oriented* case, the new user $U6$ still needs to be sent these keys individually.

It is observed that the *group-oriented* re-keying approach has the advantage over the *user* and *key-oriented* approaches, in that it requires less re-key messages.

Key-graph does not address other GKMF protocols.

7.2.3 Iolus

While Key-graph (Wong et al., 1998) (see *Section 7.2.2*) introduced the notion of secure group key management based on a hierarchy of keys, Iolus (Mittra, 1997) introduced the notion of secure group communication based on a hierarchy of nodes.

Iolus proposed a framework for securing multicast communication by addressing two scalability problems; *1 affect n* and *1 does not equal n*, where *n* is the number of group members of a multicast group.

The first scalability problem occurs when an action of one member affects the entire group. For example, from a key management perspective, when a new host joins a multicast group, an old group key may need to be replaced with a new one. If that is the case, we observe that a *join* operation exhibits the *1 affect n* scalability problem because the protocol requires all members to process the change of obtaining a new key when the new join occurs.

On similar grounds, the second problem occurs when the protocol cannot deal with the group as a whole but rather has to treat each member individually. For example, when a member leaves a multicast group, a new group key may be generated and distributed to the remaining members of the group, excluding the leaving member. Given this situation, a simple solution would be to use unicast keys (belonging to each member) to distribute the new group key securely.

Nevertheless, the basic problem is that each member must be considered individually every time a leave occurs. Thus, we observe that a *leave* operation exhibits both of the aforementioned scalability problems.

7.2.3.1 Main Architecture

Iolus addresses these problems by adopting a number of smaller multicast *subgroups.*

The main entities that are involved in the Iolus framework are:

(a) *Group security controller (GSC)*

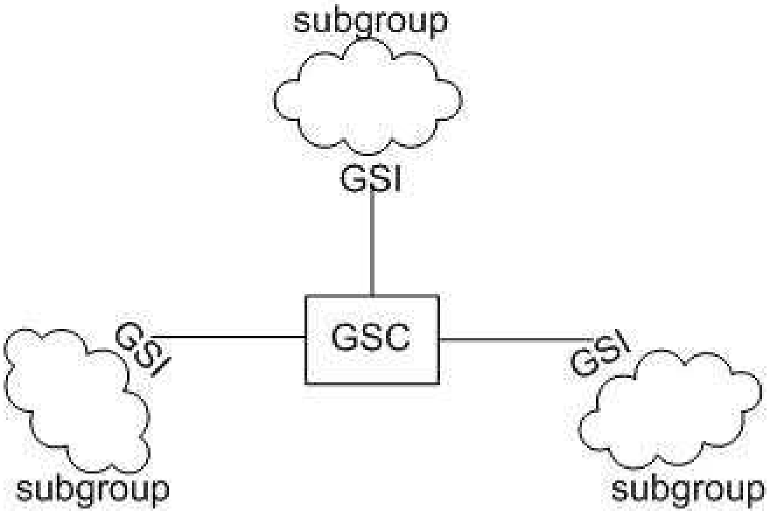


Figure 7.3: An example of hierarchy of nodes in Iolus.

GSC is the root of the hierarchy that maintains and controls the top level subgroup entities.

(b) *Group security intermediaries (GSIs)*
GSIs (also referred to as *group security agents, (GSAs)*) are the top level subgroup entities. One per subgroup, GSIs manage and connect each of the other subgroups.

We illustrate the hierarchy of nodes consisting of these main entities of Iolus in *Figure 7.3*. Each subgroup is treated independently from another subgroup in the sense that actions within a subgroup do not affect other subgroups. Iolus is thus a hybrid scheme (see *Section 4.2*) by design.

We observe that although it is not clearly addressed, it is assumed that all group members trust GSIs and GSIs trust the GSC for multicast communication.

We also assume that Iolus considers dynamic policy for its group membership.

7.2.3.2 Keys

The Iolus framework uses two symmetric keys, which are:

(a) *GSI-Member key*
This key is unique to every member of the multicast group and shared only with the member's GSI. Used for unicast communication between the GSI and the member, including secure distribution of the subgroup key to the particular member.

(b) *Subgroup key*
Primarily used for data communication, this key is unique to a subgroup and is shared by all group members within the subgroup.

7.2.3.3 Main Processes

The main process described in the framework is for establishing secure multicast data transmission, and Iolus views the whole framework as one virtual group. Iolus does not address creation of multicast groups.

For starting secure multicast, Iolus requires a GSC to initiate, which then decides who can or cannot join the group via an access control list (ACL). Once the GSC is established, other hosts such as GSIs and group members can join the group. It is assumed that the *GSI-Member key* and *Subgroup key* are established during the joining process of a particular member to the group.

Iolus limits the re-keying that needs to occur due to host joining and leaving to the subgroup level. When a new join occurs, a member joins its local subgroup. In order to protect access to previous data traffic from the newly joined member (i.e. backward secrecy), the subgroup key needs to be changed. We observe that Iolus tackles the *1 affect n* scalability problem by limiting the effect of re-keying at the subgroup level.

Similarly, when a member leaves the group, the subgroup key needs to be re-keyed, and the new key needs to be sent to the remaining group members by excluding the leaving member. As previously mentioned, *leave* can result in both the *1 affect n* and *1 does not equal n* scalability problems, because each member has to be treated separately to exclude the leaving member.

Similarly to member joining, the *1 affect n* scalability problem due to member leaving is limited to the subgroup level. We observe that Iolus does not solve the *1 does not equal n* scalability problem, but rather proposes a mitigation technique for sending the new subgroup key to the remaining members.

For this to work, Iolus uses each member's *GSI-Member key.* Instead of sending unicast messages containing the new key to every member separately, one multicast message containing n copies (assuming n remaining members) of the new subgroup key, each encrypted with a different *GSI-Member key*, is sent to the remaining member.

7.2.4 F-MSEC

Motivated by three aspects specific to multicast security; multicast application, scalability and trust relationships among entities, Hardjono et al. (Hardjono et al., 2000a) described a simple framework (referred to here as F-MSEC) for group key management from two aspects:

(a) *Network infrastructure plane*, consisting of entities and functions which define the network aspect of the framework, such as routers, hosts, as well as the routing protocols that they use.

(b) *Key management plane*, consisting of entities and functions which define the security aspect of the framework such as key managers, policy servers, as well as the security protocols that they use.

7.2.4.1 Main Architecture

The key management plane is further divided into 'regions' consisting of:

(a) *Trunk regions*
A trunk region contains only key managers, each of which is associated with at least one leaf region.

(b) *Leaf regions*
A leaf region is where all members are defined to exist. Each leaf region is associated with one key manager.

Each key management region consists of *one trunk region* and *one or more leaf region(s)*. Hardjono et al. (Hardjono et al., 2000a) claim that the purpose of introducing trunk and leaf regions is for scalability of the framework, by allowing regions to be defined according to the underlying network infrastructure, as well as the multicast applications that are under consideration. Regions can be defined to be the size of subnets or larger.

For the purpose of group key management, the main entities introduced in the framework are:

(a) *Key managers (KMs)*
Responsible for key management in trunk and leaf regions respectively.

(b) *Key translators (KTs)*
One KT for each leaf region, it translates (from encryptions under different keys) transmission payload across regions.

Figure 7.4 depicts the notion of *trunk* and *leaf* regions, along with the placement of entities in the regions.

It is assumed that all group members trust KMs and KTs for group key management.

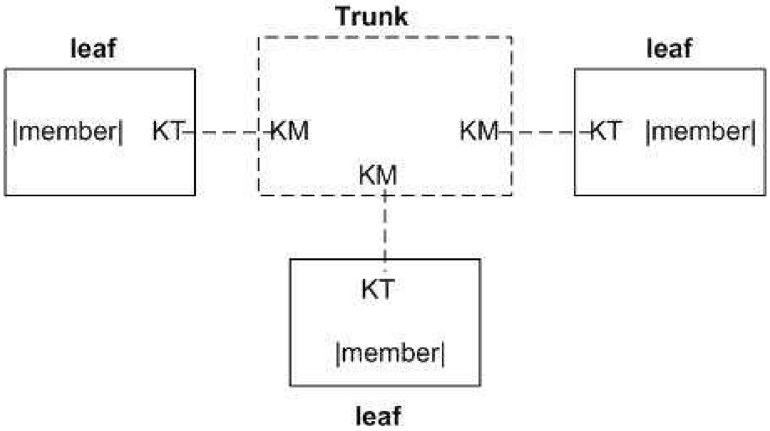


Figure 7.4: An example of notion of *trunk* and *leaf* regions in F-MSEC.

Although it is not mentioned whether F-MSEC is centralized or distributed in its design, by having KMs at each of leaf regions we observe that F-MSEC's design is based on centralized scheme.

F-MSEC briefly addresses both open and closed policy for its group membership.

7.2.4.2 Keys

In a key management region, F-MSEC introduces two symmetric keys, which are:

(a) *Trunk key*, unique for each trunk region.

(b) *Leaf key*, unique for each leaf region.

F-MSEC makes two interpretations in its view of regions, as well as from the application of cryptographic keys (the trunk and leaf keys) as follows:

(a) *Regions for delivering a group key*
Regions can act as *secure channels* for the purpose of the distribution of a group key. The group key is then used for protecting the multicast data.

(b) *Regions for delivering multicast data*
Different keys that associate with regions (trunk and leaf regions) are used to protect multicast data as it transits across regions. Thus, each region applies a different key to the multicast data, and translations of multicast data across the regions occur in the form of decryption and re-encryption.

7.2.4.3 Main Processes

We observe that F-MSEC does not clearly address any particular GKMF protocols such as creation of new groups, initial registration of keys, distribution of keys, as well as host joining and/or leaving.

However, F-MSEC briefly mentions the periodic re-keying that may occur in trunk and leaf regions, and that re-keying of trunk keys (if necessary) does not need to occur when a member at the leaf region leaves a multicast group, because trunk and leaf regions employ different keys.

7.2.5 Intra-domain GKMP

In (Hardjono et al., 2000b), Hardjono et al. proposed Intra-domain GKMP for intra-domain key management for IP multicast security. This work is based on the earlier F-MSEC (Hardjono et al., 2000a) framework.

We observe that Intra-domain GKMP extends the idea of *regions* to introduce the notion of a *domain*, where the domain is further divided into a number of smaller *areas*. The definition of domain is viewed from an administrative perspective, where a domain is administered and controlled by one body.

In order to distinguish the multicast groups for key management from the multicast group for data, Intra-domain GKMP refers to the latter as *data groups*, and the former as *control groups*. Control groups are area-wide and managed by the area key distributors. Another control group that exists at the domain level is the *All-KD-group*. This consists of all key distributors in the domain. There is only one All-KD-group in a domain.

Intra-domain GKMP places group members in one of the areas. Hardjono et al. (Hardjono et al., 2000b) claim that the purpose of placing group members in areas is to achieve flexible and efficient key management, especially when dealing with changes in group membership due to host joining (or host leaving) that may occur during the lifetime of a multicast group.

Thus, like Iolus and F-MSEC, key management operations such as generation and distribution of new keys to new members can be contained within a manageable area.

7.2.5.1 Main Architecture

The main entities that take part in Intra-domain GKMP across a domain are:

(a) *Domain key distributor (DKD)*
DKD manages and controls key management at the domain level.

(b) *Domain multicast address allocation entity (DMAAE)*
DMAAE allocates multicast addresses at the domain level.

(c) *Router/translating entity (R/TE)*
R/TE governs any multicast group that originates from outside of the domain.

(d) *Area key distributor (AKD)*
One AKD per area, which manages and controls key management at the area level.

(e) *Area multicast address allocation entity (AMAAE)*
One AMAAE per area, which allocates multicast addresses at the area level.

Trust relationships in the framework revolve around the key distributors (DKD and AKD). All group members trust these key distributors.

Although it is not clearly mentioned, we observe that Intra-domain GKMP addresses both policies for its group membership.

With the introduction of domains and areas, each of which has its own key managing entity, we note that Intra-domain GKMP is a hybrid scheme in its design.

7.2.5.2 Keys

Intra-domain GKMP uses both symmetric and asymmetric cryptography. However, for simplicity, it only uses asymmetric cryptography for key distribution (by DKD and AKDs), as well as by the multicast address allocation entities (DMAAE and AMAAE) for signing information for the multicast group members.

Intra-domain GKMP uses the following group-oriented symmetric keys:

(a) *Multicast key*
Selected by the DKD, the multicast key is unique to a multicast group. Its primary use is to secure group data traffic.

(b) *Area group key*
Selected by the AKD, the area group key is unique for each area and multicast group pair within that area. It is used for secure distribution of the multicast key to all group members of a particular multicast group in an area.

(c) *All-KD-Key*
Assigned by the DKD, this key is used to secure data traffic amongst all key distributors in the domain.

Intra-domain GKMP also uses long-term symmetric keys that are issued prior to the distribution of the group-oriented keys, which are:

(a) *Member-private-key*
Unique for each member, this key is shared with the AKD of an area and is established prior to members joining the multicast group.

(b) *AKD-Private-Key*
Unique for each AKD, this key is shared with the DKD and is established before any multicast group exists in the domain.

We observe that both of the long-term keys can be used to assist secure distribution of the group-oriented keys.

7.2.5.3 Main Processes

The main processes described in the framework are:

- *Creation of multicast groups*
  Intra-domain GKMP described two scenarios in the creation of multicast groups, each of which includes generation and distribution of multicast keys and area group keys of the new multicast group:

  (a) *Group initiation from internal-origin*
  When an initiator wishes to form a multicast group, it first initiates the creation of a new multicast group with the help of DMAAE at the domain level, which then assigns a multicast group address. The initiator then notifies its local AKD of the new multicast group, and requests a multicast key.

  The request is then passed to DKD by the AKD via a secure channel (which is encrypted by a secret AKD shares with DKD). DKD then generates the multicast key, and sends the key to every AKD in the domain.

  Upon receiving the multicast key from DKD, AKD generates and distributes its area group key to the initiator.

  Although it is not clearly mentioned, we assume that the multicast key is sent to the initiator by the AKD along with the area group key.

(b) *Group initiation from external-origin*

Group initiation from outside the domain can be done by the initiator via a border router, which then notifies the DKD of the request to create a multicast group. Alternatively, the initiator can make itself known to the AKD or the DKD directly.

Although it is not clearly mentioned, we assume that the rest of the group initiation process (i.e. generation and distribution of the multicast key and area group key) is similar to the group initiation that occurs internally.

- *Re-keying*

  Intra-domain GKMP addresses re-keying of multicast keys, area group keys, and the All-KD-Key.

  Re-keying of multicast keys is initiated and controlled by the DKD. When it is necessary to re-key the multicast key, the DKD generates and distributes a new multicast key to all AKDs in the domain, which then deliver it to members in their areas.

  Re-keying of area group keys is similar to multicast keys, except that it is initiated and controlled by the AKD of an area. We observe that area group keys are only known to the AKD and members in that area.

  Re-keying of the All-KD-Key is initiated and controlled by the DKD. When it is necessary to re-key the All-KD-Key, the DKD generates and distributes the new All-KD-Key to all AKDs in the domain.

- *Host joining*

  Intra-domain GKMP addresses two scenarios pertaining to a host joining a multicast group:

  (a) *Host joining with backward confidentiality*

  If backward confidentiality is required, re-keying of the multicast key and area group key is conducted. When a new host joins, re-keying of the multicast key is initiated by the DKD, and re-keying of the area group key is performed by the AKD.

(b) *Host joining without backward confidentiality*

If no provision of backward confidentiality is required, re-keying does not need to occur, and the newly joined member is given keys that are currently in use by the group.

- *Host leaving*

  Intra-domain GKMP addresses host leaving from the perspective that re-keying must occur to preserve forward confidentiality. Like host joining, re-keying of the multicast key and area group key (where the member leaving resides) is conducted, which results in all AKDs and group members of the multicast group (excluding the leaving member) in the domain obtaining new keys.

  Intra-domain does not consider host leaving without forward confidentiality.

7.2.6 GKM-A

This proposal (Baugher et al., 2003) is an extension of GKMP-A (see *Section 7.2.1*). GKM-A provides a common architecture for group key management protocols in multicast security that supports a variety of protocols. Each protocol can be specialized for different purposes, depending on the type of application under consideration.

GKM-A introduces the notion of a *Group Security Association* (GSA), which consists of a group of conventional security associations (SA) (which is a set of security parameters that two entities share, including encryption keys, authentication and integrity keys, as well as other attributes such as cryptographic policy of the keys and a reference index of the SA), and needs to be established prior to group communication.

GKM-A uses three types of SA to establish common keys amongst the group members, which are:

(a) *Registration protocol SA*

This SA protects the registration protocol, which occurs between the group controller and each group member.

(b) *Data protocol SA*
This SA is to protect the data communication between the communicating entities.

(c) *Re-key protocol SA*
This SA is optional because cryptographic keys may not need to be updated. The architecture also suggests that all keys could be delivered by the registration protocol.

7.2.6.1 Main Architecture

To facilitate establishment of these SAs, the main entities involved in the architecture are:

(a) *Group controller and key server (GCKS)*
A separate and logical entity that performs member authentication and authorization according to the group policy set by the group owner.

(b) *Policy infrastructure*
A separate entity that dictates group policies for multicast groups.

(c) *Authorization infrastructure*
A separate entity that provides credentials and certificates needed by any communicating entities.

We observe that GKM-A adopts a centralized scheme in its design and addresses both policies for its group membership.

7.2.6.2 Keys

In the architecture, GKM-A uses two types of symmetric key:

(a) *Key encrypting key (KEK)*
One (or more) KEKs are used to protect the TPKs and other KEKs.

(b) *Traffic protection key (TPK)*
TPK is used for securing data traffic.

Like the aforementioned SAs, these keys are generated and established during the registration protocol and/or re-key protocol.

Although not clearly addressed in the proposal, we observe that GKM-A also considers asymmetric cryptography through the use of GCKS' credential (certificate) by the group members. We assume that the credentials are issued and managed by the authorization infrastructure.

7.2.6.3 Main Processes

As we have noted, the main processes involved are:

- *Registration protocol*
  In this protocol, both entities (GCKS and group members) establish a registration SA, where entities authenticate each other, and information (such as which multicast group a member is registered to, group policy, and keys) is obtained.

  During this protocol, other SAs such as a data SA (for securing group data traffic) and a re-key SA (if necessary) are established. This includes generation and distribution of a KEK and TPK to the member.

- *Re-key protocol*
  This is an optional protocol, as dictated by a group policy in cases where keys (KEK and/or TPK) may need to be updated whenever there is a change in group membership due to new members joining and/or members leaving a multicast group, creation of new keys by GCKS, or when keys are given expiration periods.

> We observe that the GCKS is responsible for managing the re-key protocol.

Apart from these protocols, GKM-A does not clearly address other GKMF protocols such as creation of new multicast groups, new members joining, or members leaving.

7.2.7 Summary

Although the proposals (Harney and Muckenhirn, 1997), (Mittra, 1997), (Wong et al., 1998), (Hardjono et al., 2000a), (Hardjono et al., 2000b) and (Baugher et al., 2003) discussed were not intended for wireless mobile environments, many of the properties they exhibit are useful for our purpose.

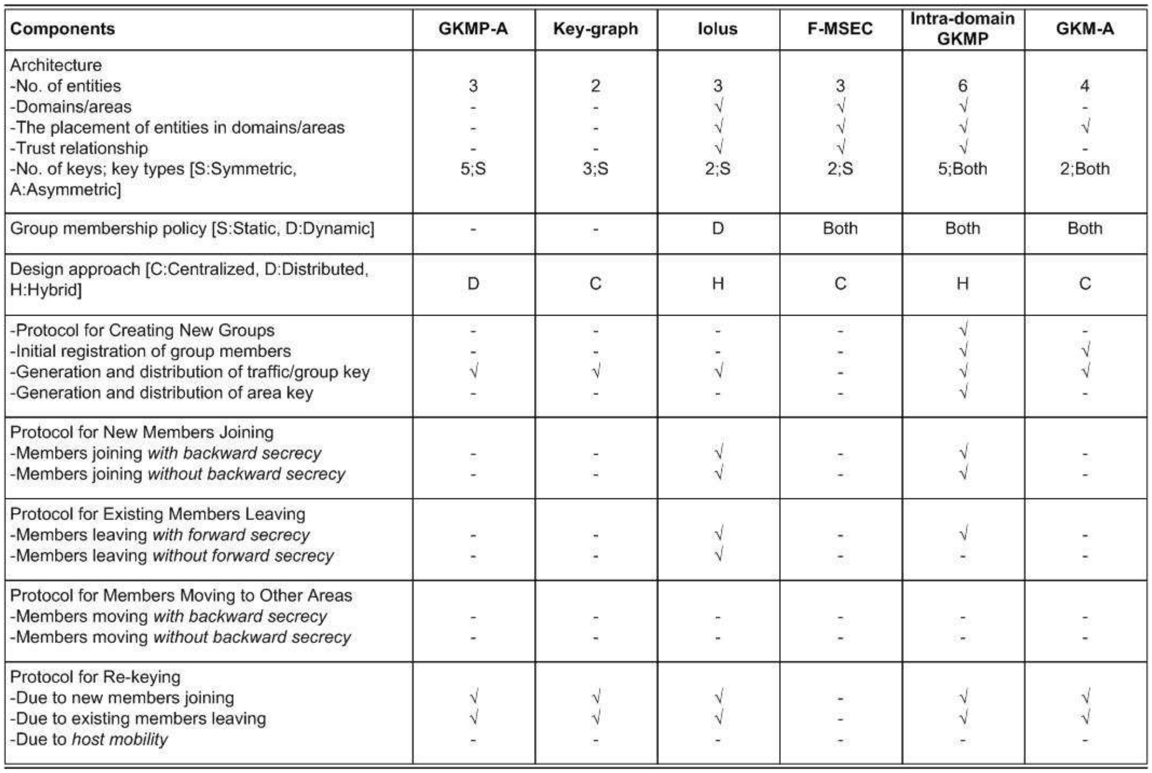
In the following section, we summarize these proposals and identify the components that are lacking. These gaps will need to be addressed in our own GKMF design.

7.3 Mapping to Generic Model for WMobEs

In this section we show how these frameworks fit into the generic model proposed in *Chapter 6*. The results of this exercise are summarized in *Table 7.1*.

Note that:

- The first column of *Table 7.1* lists the main components (including the main protocols) identified within a GKMF for WMobEs, as proposed in *Section 6.1* and *Section 6.2*.
- The other columns represent each of the discussed frameworks and give a general summary showing the extent that these frameworks fit our purpose (and show which components are lacking, or are not specified).



Components	GKMP-A	Key-graph	Iolus	F-MSEC	Intra-domain GKMP	GKM-A
Architecture						
-No. of entities	3	2	3	3	6	4
-Domains/areas	-	-	√	√	√	-
-The placement of entities in domains/areas	-	-	√	√	√	√
-Trust relationship	-	-	√	√	√	-
-No. of keys; key types [S:Symmetric, A:Asymmetric]	5;S	3;S	2;S	2;S	5;Both	2;Both
Group membership policy [S:Static, D:Dynamic]	-	-	D	Both	Both	Both
Design approach [C:Centralized, D:Distributed, H:Hybrid]	D	C	H	C	H	C
-Protocol for Creating New Groups	-	-	-	-	√	-
-Initial registration of group members	-	-	-	-	√	√
-Generation and distribution of traffic/group key	√	√	√	-	√	√
-Generation and distribution of area key	-	-	-	-	√	-
Protocol for New Members Joining						
-Members joining *with backward secrecy*	-	-	√	-	√	-
-Members joining *without backward secrecy*	-	-	√	-	√	-
Protocol for Existing Members Leaving						
-Members leaving *with forward secrecy*	-	-	√	-	√	-
-Members leaving *without forward secrecy*	-	-	√	-	-	-
Protocol for Members Moving to Other Areas						
-Members moving *with backward secrecy*	-	-	-	-	-	-
-Members moving *without backward secrecy*	-	-	-	-	-	-
Protocol for Re-keying						
-Due to new members joining	√	√	√	-	√	√
-Due to existing members leaving	√	√	√	-	√	√
-Due to *host mobility*	-	-	-	-	-	-

Table 7.1: Summary of mapping result from existing frameworks to generic model for WMobEs in *Chapter 6*.

> We use the √ notation to indicate components (or protocols) that were considered in the framework, otherwise they are indicated by a *dash*.

From *Table 7.1*, we can see that while these frameworks addressed some issues, they seem to be lacking in others. In particular:

(a) The first proposal GKMP-A does not address many aspects of GKMF, in particular host joining and host leaving protocols, which are necessary for group communication. However, GKMP-A introduced a distributed approach to managing multicast groups. The same can be said with Key-graph and F-MSEC.

(b) Unlike the others, F-MSEC addresses both policies for group membership.

(c) Iolus covers most of the components and processes necessary for a GKMF. Iolus considers desirable protocols such as members joining and leaving with the provision of backward and forward secrecy. The same can be said for Intra-domain GKMP. Neither, however, considered policies for group membership.

(d) Compared to GKMP-A, some improvements can be seen in GKM-A. However, it remains only a partial specification from our perspective.

Significantly, none of the frameworks provide components for handling *host mobility*. This is not surprising, as these frameworks were not designed with mobile environments explicitly in mind. However, providing mechanisms to address these specific problems are fundamental if a GKMF suitable for deployment in wireless mobile environments is to be fully specified.

7.4 Summary

We have looked at existing GKMFs, and showed that although some improvements have been made in more recent proposals, all lack several aspects that we have identified in our generic model, particularly issues pertaining to *host mobility*. We will use the perspectives gained from this study to influence our own specification of a GKMF for a WMobE in the next chapters.

Chapter 8

GKMF for WMobE: Scope and Requirements

In the remaining chapters, we propose our group key management framework (GKMF) for group communication in a wireless mobile environment (WMobE).

We comment on the scope of the proposal in *Section 8.1*, which represents the boundary aspects of our work. In *Section 8.2* we present the main properties and design of the framework. In *Section 8.3* we describe our proposed architecture. Finally, in *Section 8.4* we describe the main functionalities of our protocol designs.

8.1 Scope of Proposal

Regarding the scope of our GKMF specification, it is important to note the following:

- **Infrastructure-based environment**. The framework relies on an infrastructure based environment with a basic underlying cellular architecture (Bhargava et al., 2000), (Lin and Chlamtac, 2001) and (Park et al., 2002) as its networking platform. We do not intend to extend its usage to non-infrastructure environments such as wireless adhoc networks, or wireless sensor networks.

- **Group key management**. Our proposal focuses solely on the GKMF, whose main goal is to provide fundamental security support by providing all communicating entities with the necessary cryptographic keys, and providing a means to distribute these keys for the purpose of group communication. It is not the aim of the proposal to specify the details of the real data communication and how keys are used during the course of such communication.

- **Key distributions and key updates**. The aspects of key management that the framework is primarily concerned with are *key distribution* and *key updates (or, re-keying)*.

  We do not consider other aspects of key management in detail (such as the generation, storage and the disposal of cryptographic keys). Each of these is important and should be conducted in a proper and secure manner as required by the multicast application in place. We do not treat these here because they can be handled by generic techniques that are not specific to multicast group communication (see *Section 4.6* for discussion of these aspects of key management).

- **Type of Multicast Applications**. As mentioned in *Section 2.3*, multicast applications can be categorized as *one-to-many* or *many-to-many* relationships, depending on whether a single (or, many) sender(s) transmit data traffic to many receivers (group members) in the multicast group communication.

  Since the scope of the proposal is primarily concerned with key management and is not concerned with the real data communication, the type of multicast application in place does not matter and does not affect the proposal design. Therefore, our proposal does not impose any restriction on the type of multicast application in place.

- **Generic model**. The framework proposed is stated in sufficient abstraction that it can easily be made compatible with existing network protocols, as well as application-layer security protocols to allow for practical implementation for group communication in WMobEs. In order to further support this, we suggest the use of mechanisms and techniques that are based on standards (such ISO/IEC and Internet Standards).

8.2 Properties and Design Requirements

In this section, we present the main properties and design requirements necessary for our proposal. We divide the requirements into three categories; general requirements, security and trust requirements, and performance requirements. These requirements follow from discussions held in *Chapter 4* and *Chapter 5*.

Each of these is presented in the following subsections.

8.2.1 General Requirements

As discussed in *Section 4.3*, *Section 4.7.3*, and *Section 5.3*, we list the general requirements necessary for our proposal, as follows:

- *Dynamic group membership policy.* Group members should be allowed to join and/or leave a multicast group at any time.
- *Provision for host mobility.* A protocol should be provided for allowing group members to move to other areas while still remaining in a group session.
- *Scalability.* A GKMF should be scalable enough to expand its group size over time.
- *Reliable and trustworthy key manager.* All key managers managing the cryptographic keys and governing group operations should be trusted by group members.

8.2.2 Security and Trust Requirements

As discussed in *Section 4.3*, *Section 4.5*, and *Section 5.3*, we list the main security and trust requirements for our proposal:

- *Entity authentication.* Both group members and key manager entities should be able to authenticate and verify each other's identities.
- *Backward and forward Secrecy.* From the aspect of key management, rekeying needs to occur whenever there is a change in group membership (either due to new joins, member leaves, or member moves) for provision of backward and forward secrecy.
- *Data (Message) integrity and authentication.* Both group members and key manager(s) should be able to verify the integrity of data received.
- *Secure data exchange.* Communications amongst group members should be protected (and remained confidential), where authorized access is given only to group members.
- *Key distribution.* All necessary keys should be securely distributed to all group members prior to group communication.
- *Key updates (re-keying).* Key updates should be done in a secure and proper manner. Group members should be informed when an update of keys is required.
- *Additional key management during host mobility.* Generation, distribution and re-keying of new cryptographic keys may be required to support host mobility.
- *Trust model.* A trust model is crucial for a GKMF in order for it to properly function. This includes a security architecture that is compatible with existing network protocols, scalable and transparent to higher level applications and services.

8.2.3 Performance Requirements

As discussed in *Section 5.2* and *Section 5.3.3*, we list the performance requirements necessary for our proposal:

- *Computation, energy and storage requirements.* Many mobile devices have limited processing power, as well as storage capacity, and these should be kept to minimal. This includes requirements for:

  (a) keeping the number of encryptions (or decryptions) involved to a minimum.

  (b) using encryption (or decryption) techniques that are not computationally intensive.

  (c) keeping the number of keys that a member (device) needs to store to a minimum.

- *Communication and bandwidth requirements.* As wireless mobile networks are susceptible to frequent disconnection, and available bandwidth is limited, the communication overhead between group entities should be kept to a minimum.

8.3 Main Architecture

In this section, we propose the architecture that we will use for our framework. We use the main structural components identified in *Section 6.1.1* to describe the architecture in detail.

We first determine the aspects that influenced our design decision.

8.3.1 Design Influence

In this section, we review components from the existing architectures in *Chapter 7* that we adopt in order to satisfy the design requirements of *Section 8.2*.

(a) *Domains and Areas*

We will adopt the notion of *domains* and *areas* (see *Section 7.2.5*) as the main structural components in the framework architecture. This idea facilitates scalable and efficient distribution of keys to all group members, as group members are defined to exist in individual areas that are locally managed by a trusted entity.

We will extend these ideas by introducing *inter-domain* relationships for group communications that originate from other domains. This was not considered in existing proposals. See *Section 8.3.3.1* for more details.

(b) *Subgroups*

By placing group members in individual areas, we can associate them with the concept of subgroups (see *Section 7.2.3*). In other words, group members are *subgrouped* in particular areas. By doing so, we seek to overcome scalability problems that may occur whenever there is a change in group membership. When a new member joins (or an existing member leaves) a multicast group, it joins (or leaves) its local area and does not affect the other *subgroups* (in other areas) in the domain.

Host mobility between areas can also be managed in a more efficient way.

(c) *Symmetric cryptography*

We follow previous GKMF proposals, and adopt symmetric cryptography in our proposal. This is primarily due to reasons pertaining to the nature of the wireless mobile environments (see *Section 5.2*) that our framework is intended for. See *Section 8.3.6* for more details.

(d) *Key hierarchies*

Hierarchies of keys are very useful for group communication in WMobEs where group members may move between areas that may have their own security requirements. We will adopt the concept of key hierarchies in our framework (see *Section 8.3.6*).

In the following sections, we describe how each of these fits into our GKMF.

8.3.2 Main Entities and Their Roles

In this section, we present the main entities needed in the architecture.

The main controlling entities in both domain and area(s)(see *Section 8.3.3*) are the following:

(a) *Domain key manager (DKM)*
At the domain level, a DKM is defined to exist, whose main responsibility is generating, distributing, storing and deleting all keying materials that may be required.

We also assume that the DKM plays the role of group controller, which includes managing group policies, group membership, re-keying events and security policies.

However, in practice, the domain key manager and group controller can be separate entities, as observed in (Harney and Muckenhirn, 1997), (Mittra, 1997), (Hardjono et al., 2000b) and (Baugher et al., 2003). We assume that there is only one DKM per domain, and that it is the main reference for security parameters for other key managers in the domain. In collaboration with other key managers, the DKM governs host mobility across the domain.

In summary, the DKM's main roles are:

- main key manager of a domain,
- collaborating with other key managers (at the area level) to provide secure and efficient key management services within a domain,
- generating and distributing cryptographic keys to all area key managers in the domain,
- governing all re-keying events that may occur during the lifetime of a multicast group,
- working closely with area key managers to govern host mobility.

(b) *Area key manager (AKM)*
One AKM is defined for each area. The main responsibility of an AKM is running the key management aspects relating to an area, including those of the group members residing within that area. Operating under the DKM's jurisdiction, an AKM is responsible for any re-keying event that

may occur at the area level. The AKM also works closely with the DKM to manage host mobility that may occur across the domain.

In summary, the AKM's main roles are:

- main key manager of an area,
- assisting the DKM to provide secure and efficient key management services to group members in areas,
- generating and distributing cryptographic keys to all group members residing in an area,
- governing re-keying events at the area level, operating under the DKM's jurisdiction,
- working closely with the DKM and other AKMs to govern host mobility.

(c) *Group members (M)*
From a group communication perspective, group members consist of sender(s) and receiver(s). A group member is defined to reside within one area at any given time.

8.3.3 Domain(s) and Area(s)

In this section, we look more closely at the domain(s) and area(s) within the architecture. We also discuss multiple domain relationships, in the cases where inter-domain group communication is permitted.

The notion of domain(s) and area(s) provides a means to have an administratively manageable environment for group communication to take place, most importantly for efficient key management.

In our proposal, a *domain* can be logically or physically defined. Either way, it is controlled and managed by a trusted entity operating under one system, for instance the *Global System for Mobile Communications* (GSM) operator's network (Lin and Chlamtac, 2001).

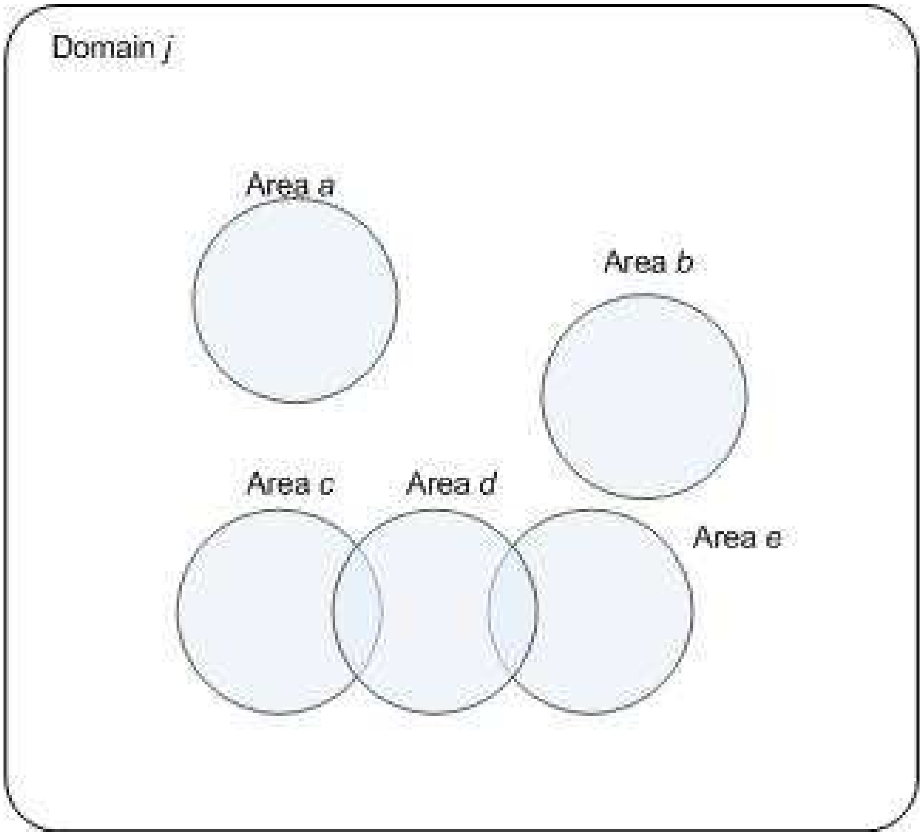


Figure 8.1: An example showing the notion of domain and area.

The domain is further divided into a number of smaller manageable areas, each of which is managed by an AKM, operating closely with the DKM with regards to key management. We illustrate the notion of domain and areas in *Figure 8.1* where, Domain *j* is further divided into several areas labeled Area *a* to *e*, each of which can be logically or physically overlapping with one another.

Since a domain is controlled by one DKM, all corresponding entities across a domain should be able to interface successfully with one another. Although areas within a domain may be using similar systems for interoperability, this does not change the fact that each area is unique and that a group member that moves from its local area to another area must obtain security information (i.e. keys) associated with that area prior to, or during, the move.

In addition, different areas may contain security information that is meant only for group members residing in that area, hence access control must be in place to protect the local information. Furthermore, in WMobEs host mobility may place group members in different areas, each of which has its own restriction on what information may or may not be accessed by the mobile members.

We use the following definitions to differentiate areas during group operations:

(a) *Local areas*
The term *local area* is used to refer to the area where hosts (potential group members) first join a multicast group.

(b) *Visited areas*
The term *visited area* is used to refer to other areas in a domain, where group members may or may not *move to* (during host mobility) throughout the lifetime of their group membership.

8.3.3.1 Inter-domain Relationship

In this section, we introduce the idea of *inter-domain* relationships. This is useful in cases where group operations originating from outside the local domain are permitted (for example, when a host or potential member wishes to join a multicast group that is managed by other domain). This kind of request is called a *cross-domain request*.

There are two approaches that can be adopted in dealing with inter-domain communication:

(a) *Use of an intermediate entity*
The first approach is to have a separate entity, which can be in the form of a server or a router, to deal with any inter-domain communication (if it occurs). For example, any request to join a multicast group that is not in the current domain where the request originates from is transferred to the aforementioned entity, which deals with the inter-domain requests. This entity thus acts as an intermediate node or a bridge between the two

distinctive domains. An intermediate host may cater for two or more inter-domains communications depending on its hardware or software capacity.

This intermediate entity is similar to existing proposals in (Hardjono et al., 2000a) and (Hardjono et al., 2000b). Briefly, (Hardjono et al., 2000a) and (Hardjono et al., 2000b) discuss the need for a translation entity or a router that is able to translate any cryptographic messages protected by foreign keys that are unintelligible to the current domain.

(b) *Cross-referencing between DKMs*

An alternative approach is the cross-referencing between DKMs. In this case, DKMs from both affected domains are the ones governing the inter-domain requests.

For example, let D_i and D_j denote domain i and domain j. Any host who wishes to join a multicast group outside its local domain D_i (the point of where it is residing at the time of the request), is managed by its domain key manager (DKM). The DKM then liaises with the DKM in D_j to govern the request, including any security relationship exchange that may occur during the course of the host request. Both DKMs need to collaborate in order to realize inter-domain communication.

Cross-referencing between DKMs avoids the need for an additional entity or node, and is similar to the inter-Base Station handoff concept of IS-41 (Lin and Chlamtac, 2001), in which base stations under the same controlling entity collaborate to govern handoff for members that move from one area of a base station to another area of another base station.

8.3.4 Placement of Entities

In this section, we place the entities specified in *Section 8.3.2* in the domains and areas specified in *Section 8.3.3*.

As mentioned in *Section 8.3.2*, a DKM oversees key management at the domain level, and an AKM oversees key management at the area level. We illustrate placement of entities in two instances in *Figure 8.2* and *Figure 8.3*.

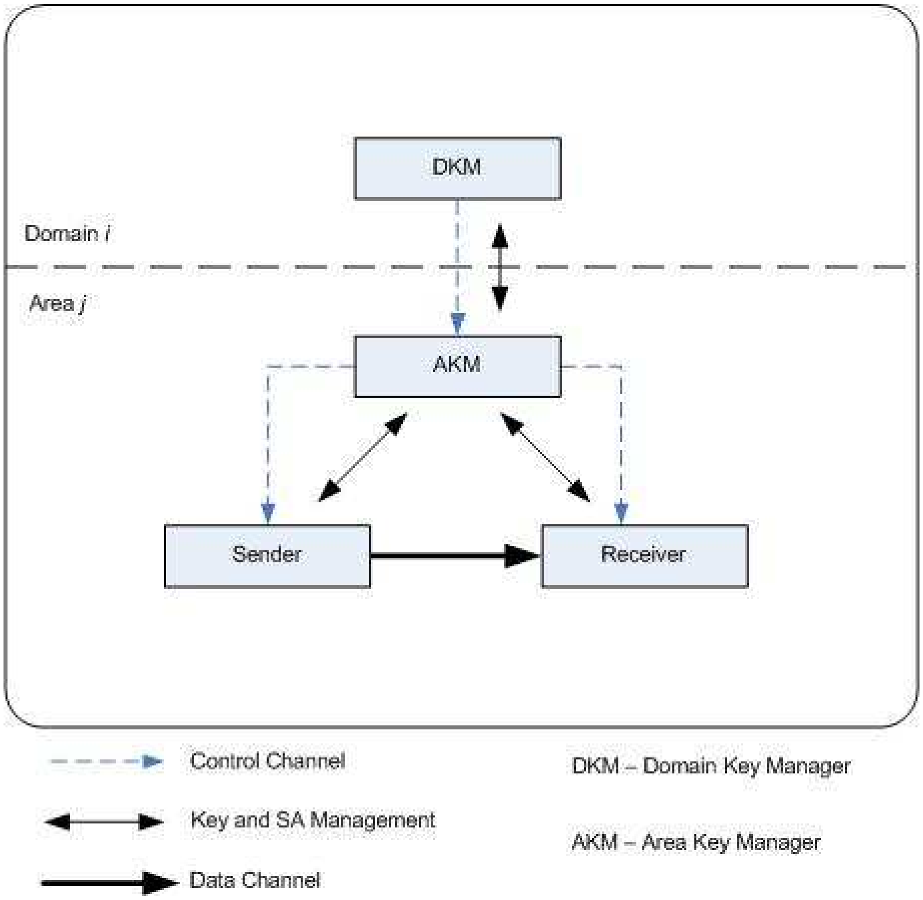


Figure 8.2: Placement of entities in domain i and area j.

Figure 8.2 shows placement of entities in domain i and area j. From the illustration, while DKM is the main key manager of domain i and AKM is the key manager of the area j, we assume there is a sender and a receiver to represent the group members of the multicast group in place.

The horizontal dotted-line shows a logical division between domain i and area j. The *dotted-arrow* lines from DKM to AKM, as well as from AKM to both sender and receiver, show *control channels* which can be used by the DKM and AKM for transmitting control messages, such as notification on re-keying that has taken place, or acknowledgement of messages received.

While *double-arrow* lines show the exchange of key and SA management between pairs of entities, the *single-arrow* line from sender to receiver shows the data channel of real group communication which may take place after the

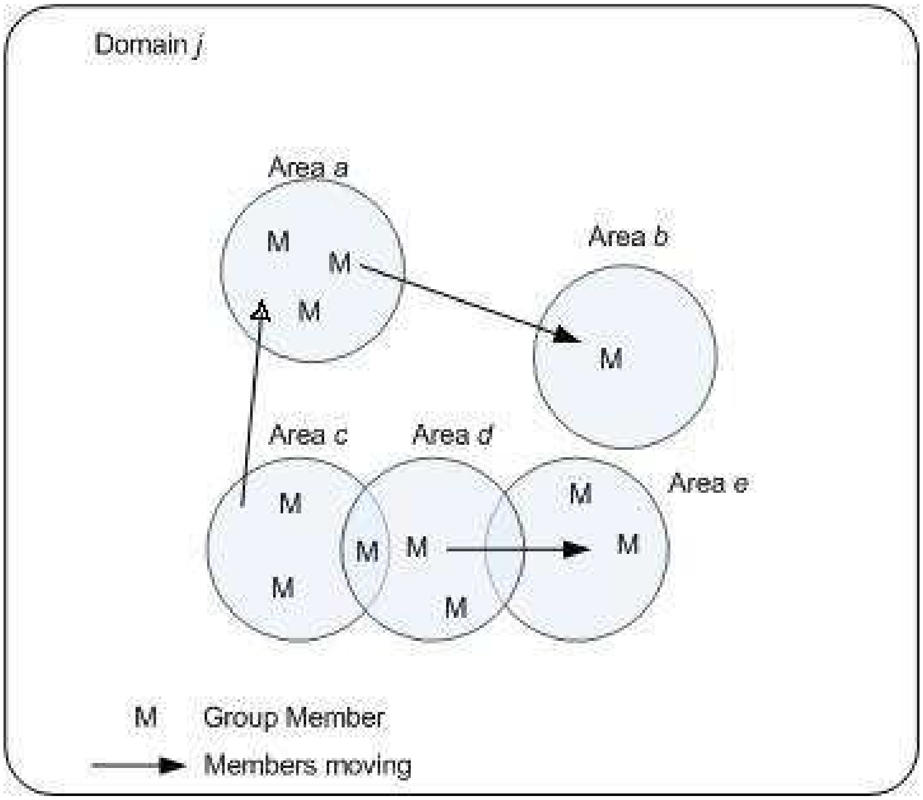


Figure 8.3: Placement of group members throughout areas.

exchange of key and SA management occurs.

On the other hand, *Figure 8.3* shows placement of group members M across a domain j, where distribution of members occurs throughout the areas a to e. The *arrows* denote the movement of group members between the areas.

8.3.5 Trust Relationships

As mentioned in *Section 6.1*, trust relationships often revolve around key managers, who are the main key distributors. In our framework, we assume that all key managers (DKM and AKMs) in a domain are trustworthy and reliable. All group members of multicast groups trust these key managers, particularly for providing secure key management services (see *Section 4.6*).

There are two levels of trust relationships:

(a) *At the domain level*
At the domain level, all AKMs trust the DKM as the primary key distributor, as well as the main group manager for various multicast groups operating in that domain.

(b) *At the area level*
At the area level, all group members (residing in that area) trust their AKM as the main reference point for security parameters needed for group communication.

8.3.6 Types of Key

In this section, we look at another of the fundamental components of the proposed architecture, namely the cryptographic keys.

We base our framework on symmetric key cryptography because most mobile devices that operate in WMobEs exhibit special characteristics (see *Section 5.2*), which benefit from the computationally faster and less complex techniques offered by the symmetric approach.

In the following sections, we look at the symmetric keys used in our framework, which are categorized into two main groups; *long-term* and *short-term* secret keys. We assume that all symmetric keys used are of an appropriate recommended length (at the time of writing we recommend 128 bits), allowing the use of standard algorithm such as AES (FIPS, 2001).

We end this section by looking at the aspects of key management that we assume are in place and available for the purpose of group key management, and that we will thus not fully specify in the framework.

8.3.6.1 Long-term Keys

Long-term keys are assumed to have been established prior to any host joining a multicast group.

Entities in the framework use these keys to securely initiate secure group communications, including when disseminating short-term keys.

There are three types of long-terms key in the framework; *Domain-Area keys, Domain keys* and *Area-Member keys.* The following subsections describe each key in turn, and their details are summarized in Table 8.1.

- **Domain-Area Key,** DA_i_Key
  The Domain-Area key is the unique long-term key shared between the DKM and a specific AKM in a domain. More precisely, DA_i_Key corresponds to the symmetric key shared between DKM and the area key manager AKM$_i$ of area i.

  This unique key is established with every AKM in the domain prior to any request to create multicast groups in the domain. Each key is generated and distributed by the DKM to every AKM by an appropriate secure means (see *Section 8.3.6.3*).

  We assume that the membership of all key managers in a domain is predetermined and fixed (see *Section 3.2.1*). Thus, each Domain-Area key is static and valid until the policy determines otherwise. Once a Domain-Area key expires (or is revoked), a new key must be generated and distributed to the affected AKM.

  The function of each Domain-Area key is restricted only to *unicast* communication between the DKM and a particular AKM.

- **Domain Key,** D_Key
  The Domain key D_Key is the long-term key shared by all key managers (i.e. DKM and all AKMs) in a domain. Like the Domain-Area key, the domain key is established prior to any request to create multicast groups in the domain. D_Key is generated and distributed by the DKM to all

AKMs via secure channels. One such channel is created by unicasting under the appropriate *Domain-Area* keys.

Since the group membership of all key managers is static, D_Key is also fixed and valid until the policy determines otherwise. A new domain key must be generated and distributed to all AKMs to replace the old one on its expiry.

Although D_Key is often used to assist secure distribution of other keys needed for group communication, it is also referred to as a *control key* because it can be used by the DKM to send control messages, such as to notify all AKMs in the domain of a new multicast group, re-keying events that are taking place, as well as any notification concerning host mobility.

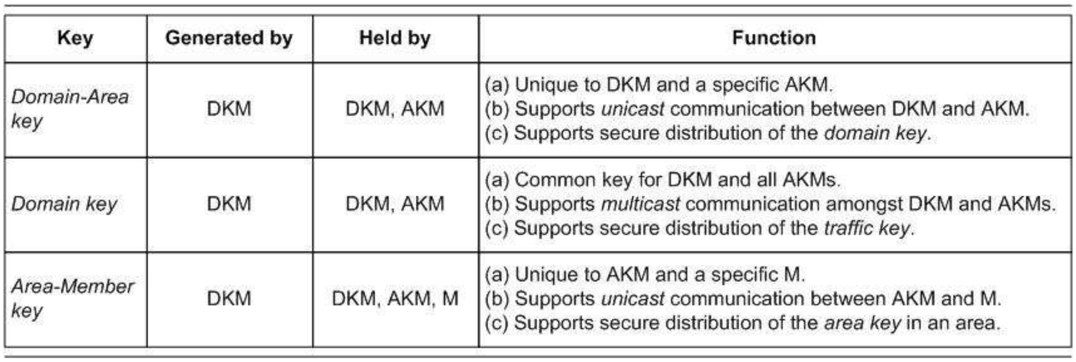
A desirable function of D_Key is that it provides a means for *multicast* transmission of messages among all key managers in a domain.

- **Area-Member Key,** A_iM_Key
  Another long-term key used is a unique Area-Member key shared between the AKM of an area and every group member residing in that particular area. More precisely, A_iM_Key corresponds to a symmetric key shared between area key manager AKM$_i$ and a group member M.

  This key is obtained during the first contact that a host (who soon will become a member of a multicast group) makes to the AKM of a particular area to register with a particular multicast group.

  We assume that the Area-Member key is established prior to any request to join the group, and is generated and sent by the DKM to a particular AKM, which then sends it to the group member M. More precisely, at the domain level, the DKM uses the Domain-Area key to secure the distribution of the Area-Member key to a particular AKM of an area. Then, at the area level an AKM uses a *secure means* (see *Section 8.3.6.3*) to distribute the key to the group member M.

  Since group membership can be dynamic, we assume that the Area-Member key remains valid throughout the lifetime of a particular multicast group, or until the group member has ceased to be a member of that particular multicast group.



Key	Generated by	Held by	Function
Domain-Area key	DKM	DKM, AKM	(a) Unique to DKM and a specific AKM. (b) Supports *unicast* communication between DKM and AKM. (c) Supports secure distribution of the *domain key*.
Domain key	DKM	DKM, AKM	(a) Common key for DKM and all AKMs. (b) Supports *multicast* communication amongst DKM and AKMs. (c) Supports secure distribution of the *traffic key*.
Area-Member key	DKM	DKM, AKM, M	(a) Unique to AKM and a specific M. (b) Supports *unicast* communication between AKM and M. (c) Supports secure distribution of the *area key* in an area.

Table 8.1: Summary of long-term keys and their functions.

Like the Domain-Area key, the function of each Area-Member key is restricted only to *unicast* communication between the AKM of an area and a group member of that area.

8.3.6.2 Short-term Keys

Short-term symmetric keys are assumed to have been established after group members join multicast groups (or after the long-term secrets have been established). As group members are already in possession of long-term keys, these are often used to assist secure distribution of short-terms keys.

There are three short-terms keys introduced in the framework; *traffic encryption keys, area keys* and *session mobility keys.* The following sections describe each type of key in turn, and at the end their details are summarized in Table 8.2.

- **Traffic Encryption Key,** T_Key

  The Traffic Encryption Key (or *traffic key*) T_Key is a short-term key shared by all group members of a particular multicast group in a domain. There is a unique traffic key for a specific multicast group. T_Key is generated and distributed by the DKM to all AKMs in the domain, which then in turn disseminate this key to all group members residing in their area. The establishment of the key takes place only after the first host creates (and joins) a multicast group.

There are several options for securing the distribution of the traffic key to all group members in the domain. One option is for the DKM to use *unicast*, by protecting the key under each Domain-Area key and thus sending it to every AKM independently. At the area level, an AKM uses the same method using each Area-Member key.

Another option is for the DKM to use *multicast* and send a single message containing the traffic key to all AKMs protected under the domain key D_Key. Each AKM in turn distributes the traffic key to all group members in the area. If all group members residing in the area belong to the same multicast group, AKM can send the traffic key by multicast using the area key A_Key (see next subsection). Otherwise, AKM will have to unicast to every member protected under the Area-Member key.

The main function of the traffic key is to protect the real data in communication. The traffic key is valid throughout the lifetime of a multicast group, or until the policy determines otherwise. To replace a traffic key, a new traffic key must be generated and distributed to all group members in the domain.

- **Area Key,** A_Key

  The Area key is a short-term key unique to an area. Every area has a different A_Key. An area key is generated and distributed by the AKM of a particular area, and shared only by group members residing in that area.

  An area key is established with a group member after it joins a multicast group. The dissemination of the area key to all group members is done by the AKM via *unicast* methods using each Area-Member key.

  The main purpose of having an area key, which is unique to an area, is:

  (a) *To securely manage host mobility across areas in the domain*
  Without proper control, group members that are moving from one area to another may collect security information, including old keys not authorized to them. In addition, different areas may have different access control pertaining to their local information. By having

adequate access control, one can prevent such security violations by unauthorized moving members.

(b) *To provide efficient and scalable re-keying*
Unique area keys are useful for efficient re-keying and promote scalability, since group members within an area are managed under one unique key.

Note that while group members of a multicast group can be dispersed across the domain due to host mobility, each group member may have in his possession a different set of area keys. The first area key that a host may possess is from the area where it resides the first time it joins a multicast group, in other words the *local area* (see *Section 8.3.3*).

An area key is often used to assist secure multicast distribution of the traffic key to all group members in an area in a single transmission.

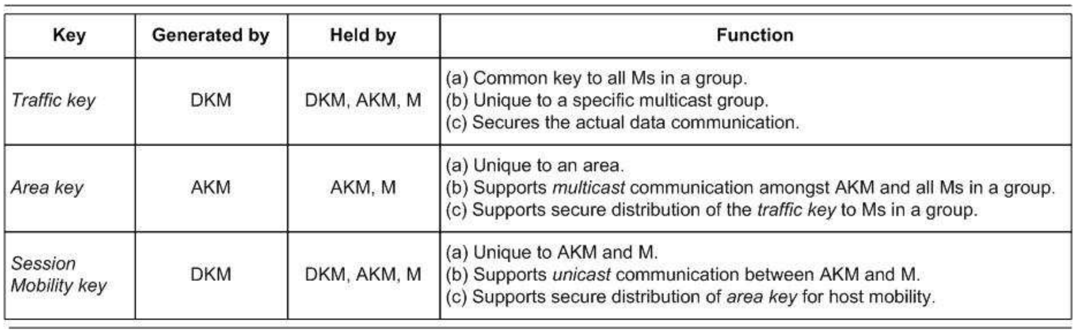
An area key is assumed valid as long as there are members residing in that area, or until the policy determines otherwise.

- **Session Mobility Key,** $S_{m-}Key_{iv}$

The Session mobility key $S_{m-}Key_{iv}$ is a short-term symmetric key shared between an AKM and a moving group member. More precisely, $S_{m-}Key_{iv}$ is a session mobility key shared between an area key manager AKM$_v$ and a group member M_i. This key is only used for host mobility and exchanged during the hand-off operation.

This key is established between a moving member and the AKM of a visited area prior to host mobility. The generation and initial distribution of this key is conducted by the DKM in a domain, then delivered to the group member via an AKM in the local area (where the member is currently residing). The same key is then delivered to the AKM of a visited area by the DKM. The delivery of the session mobility key must be done using secure channels (see *Section 8.3.6.4*).

This key is used for any *unicast* communication that may occur between the AKM of the visited area and the mobile member throughout its residence period in that area. This includes the secure distribution of the visited area's area key to the mobile member.



Key	Generated by	Held by	Function
Traffic key	DKM	DKM, AKM, M	(a) Common key to all Ms in a group. (b) Unique to a specific multicast group. (c) Secures the actual data communication.
Area key	AKM	AKM, M	(a) Unique to an area. (b) Supports *multicast* communication amongst AKM and all Ms in a group. (c) Supports secure distribution of the *traffic key* to Ms in a group.
Session Mobility key	DKM	DKM, AKM, M	(a) Unique to AKM and M. (b) Supports *unicast* communication between AKM and M. (c) Supports secure distribution of *area key* for host mobility.

Table 8.2: Summary of short-term keys and their functions.

In the case where a mobile member is moving to another area from its visited area, it then has to establish another session mobility key with another area key manager of the area it is moving into. Unless policy determines otherwise, a group member may possess a session mobility key for every area that it visits throughout the lifetime of its group membership. This session mobility key is valid throughout the member's residing period in that area or until it ceases to be a member of the multicast group.

The function of the session mobility key is restricted only to *unicast* communication between the AKM and a particular group member of a multicast group.

8.3.6.3 Assumptions on Aspects of Key Management

As mentioned in *Section 8.1*, we will focus on distribution and updating of (mainly short-term) cryptographic keys. Other operations are not treated here in detail because their provision can be achieved by generic solutions which are not specific to group communication. The following assumptions regarding key management operations are made:

(a) Key generation is conducted in a secure and proper manner by all key managers (DKM and AKMs) in a domain. We assume that key managers use recognized key generators to generate keys as randomly as possible.

(b) Long-term key distribution is conducted in a secure manner, prior to any group being established. These keys can be established using various key establishment methods in ISO/IEC 8732 (ISO, 1988), ISO/IEC 11770-1 (ISO, 1996a) and ISO/IEC 11770-2 (ISO, 1996b). There are several accepted methods that can be used to distribute keys to the communicating entities, including via physical (manual) delivery techniques, using other key to encrypt keys (key encrypting keys), or using a trusted third party.

(c) Key storage is managed in environments equipped with secure technology. For example, tamper-resistant hardware can be used to increase the level of security of the stored keys.

(d) Key installation is performed securely by all key managers (DKM and AKMs). We assume that a DKM is responsible for key installation at the domain level, and an AKM is responsible for key installation at the area level.

(e) Key revocation is conducted in a secure and proper manner by all key managers in a domain. We assume that a DKM governs any process to revoke keys at the domain level, and likewise an AKM at the area level.

(f) Key disposal is handled in a secure and proper manner by all key managers such that no other information can be used to recover the disposed keys. We assume that DKMs and AKMs manage key disposal processes.

8.3.6.4 Secure Channels

We use the term *secure channel* to mean that communication between group entities (key managers and group members) within the GKMF is protected by careful application of symmetric keys. This is achieved as follows:

(a) Key managers (DKM and AKMs) at the domain level secure the communications between them by using either a common key such as *domain* *key* for protecting communications between all key managers, or *Domain-Area* keys for secure communications between the DKM and each AKM separately.

(b) Area key managers and group members in the same area secure communications either by using a common *area key* for protecting communications between the AKM and all group members (residing within that area), *Area-Member* keys between the AKM and a group member separately, or *session mobility* keys between an AKM and a mobile member.

Secure channels are created when group entities (DKM, AKMs and group members) use these keys in the course of group communication.

8.3.7 Group Membership Policy

In our proposal, while dynamic group membership (see *Section 3.2.1*) is assumed throughout the framework design, the option for static group membership is also made available.

8.3.8 Design Approach

As mentioned in *Section 4.2*, the assignment of key manager(s) can be *centralized*, *distributed*, or *hybrid*. Our proposal adopts the hybrid approach in the assignment of key managers, and is based on a distributed hierarchy of trusted entities (DKM and AKMs) for key management.

8.4 Protocol Functionalities

In this section, based on the main protocols identified in the generic model in *Section 6.2*, we describe the required functionalities of each protocol in our GKMF. The protocols themselves are specified in *Chapter 9*.

8.4.1 Creating New Group and Initial Distribution of Keys

This protocol governs the creation of a multicast group and describes the initial key distribution to all key managers and the first host (group member) to create a multicast group.

The main functional requirements of this protocol are to:

- create a multicast group for the purpose of group communication.
- distribute a traffic key T_Key by the domain key manager to all area key managers.
- distribute a traffic key T_Key and an area key A_Key by the area key manager (of an area where the host residing) to the first host to create a multicast group.

The main security requirements of this protocol are to ensure that:

- only an authorized host is allowed to create a multicast group.
- communication between the domain key manager and area key managers is secure.
- communication between the area key manager and the host (or group initiator) is secure.
- the distribution of the traffic key T_Key and the area key A_Key to the area key manager and the host is protected.

The main information disseminated during this protocol includes:

- *Group membership policy*
  The type of group membership policy of a multicast group is indicated by a *1-bit* flag value. A 0 flag indicates that the group membership policy is

static, while a 1 flag indicates that the policy is *dynamic*. This flag value is depicted in the protocol message contents as *gm* (*group membership*).

- *Backward and forward secrecy*
  The need for these confidentiality services is indicated by a *2-bit binary* flag value, as shown in *Table 8.3*, where these services are presented as *on* for a *required* service, or *off* for a *not required* service.

Flag value	**Confidentiality Service**		**Security Requirement**
	Backward Secrecy	*Forward Secrecy*	
00	Off	Off	None
01	Off	On	Only forward secrecy
10	On	Off	Only backward secrecy
11	On	On	Both

Table 8.3: Requirements for backward and forward secrecy.

This flag value is indicated in the protocol message contents as *conR* (*confidentiality requirement*).

Other related information, such as the lifetime of each key and the type of algorithm, are also distributed during this protocol. However, we do not explicitly address these here and assume that this information exists in the *text* field of the protocol message (see *Section 9.1.5*).

8.4.2 New Member Joining

For a group communication to take place, a multicast group should have at least two (or more) entities in place. This protocol governs the *join* (or registration) of group members into the multicast group. As discussed in *Section 6.2.2*, when a host wishes to join a multicast group, two approaches can be adopted based on whether providing backward secrecy is necessary or not. If providing backward secrecy is required then any new host that wishes to join the group must not be allowed to access previous traffic prior to its admission to the group. Otherwise, no such restriction applies. Thus, this protocol will consider both options of hosts joining without and with backward secrecy.

In the first option, any host that joins the group is given the same set of keys that are currently used by the group members, and the protocol does not need to update (or re-key) the current keys. On the other hand, the second option requires the protocol to re-key the group with a new set of keys. Similarly to the first join to the multicast group by the group initiator, subsequent group members also receive the traffic key T_Key and the area key A_Key along with other information associated with the multicast group joined (see *Section 8.4.1*). In the event that a join request originates from outside the current domain, the request is managed by a governing entity in place (see *Section 8.3.3.1*).

The main functional requirements of this protocol are categorized in two parts, corresponding to the aforementioned *join* options:

(a) For host joining *without* backward secrecy, the main functional requirements are to:

- add new group members into a multicast group.
- deliver a set of keys (which are traffic key T_Key and area key A_Key) to the newly joined member.

(b) For host joining *with* backward secrecy, in addition to those listed in (a), other functional requirements are to:

- initiate a re-keying of a traffic key T_Key in the domain.
- initiate a re-keying of an area key A_Key in the area (where the join occurs).
- deliver a new set of keys (which are a new traffic key T_Key_{new} and a new area key A_Key_{new}) to the newly joined member.

The main security requirements of this protocol are to ensure that:

- only an authorized host is allowed to join a multicast group.
- communications between the domain key manager and the area key manager are secure.

- communications between the area key manager and the requesting host are secure.
- for member joining without backward secrecy, the distribution of the traffic key T_Key and the area key A_Key to the newly joined member is protected.
- for member joining with backward secrecy, the distribution of the new traffic key T_Key_{new} and the new area key A_Key_{new} to all area key managers and all group members (including the newly joined member) in a domain is protected.

8.4.3 Existing Member Leaving

Dynamic environments allow group members to leave multicast groups at anytime. This protocol governs the leaves (or de-registration) of a group member from a multicast group. As discussed in *Section 6.2.3*, two options can be adopted depending on whether the provision of forward secrecy is necessary, or not. If it is, then past group members (including the ones who are leaving) must not be allowed to access future communications. Otherwise, no such restriction applies. This protocol will consider both options of members leaving with and without forward secrecy.

Unlike members joining, leave cases can be *voluntary*, or *involuntary* (see *Section 4.3.2*). Involuntary leaves in particular may require protection from the ejected members (in such cases forward secrecy is necessary).

As in members joining, while members leaving without forward secrecy will not affect the current group keys (no updates of keys are necessary), members leaving with forward secrecy require the protocol to re-key the group with a new set of keys. In the event that a leave notification originates from outside the current domain, the request is managed accordingly by the governing entity in place (see *Section 8.3.3.1*).

If it is deemed necessary for key managers (such as a DKM) to keep track of the group members who leave a multicast group (regardless of whether it occurs due to voluntary or involuntary reasons), a key manager will need to maintain a table containing information on the leaving members. The key manager may need this information for future reference should the same members wish to later re-join a multicast group. This list is referred to as *HisList*, which stands for *History List* (see *Section 9.1.4*).

The main functional requirements of this protocol depend on the aforementioned *leave* options, and are listed as follows:

(a) For members leaving *without* forward secrecy, the main functional requirement is to remove existing group members from a multicast group.

(b) For members leaving *with* forward secrecy, in addition to those listed in (a), the other main functional requirements are to:

   - initiate a re-keying of a traffic key T_Key in the domain.
   - initiate a re-keying of an area key A_Key in the area where the leave occurs, as well as in area(s) visited by the leaving member.

The main security requirements of this protocol are to ensure that:

- only leave notifications coming from authorized entities (such as group members or a domain key manager) are processed.
- communications between the domain key manager and the area key manager are secure.
- communications between the area key manager and the group member are secure.

8.4.4 Member Moving to Other Areas

In dynamic wireless mobile environments, group members are not just allowed to join (and/or leave) a multicast group, but are also allowed to move between areas while remaining in a group session. As each area may have different security requirements, as well as keeps its own security information (such as old keys), members moving to other areas can be viewed as new joins, each of which may require protection of its local information from an entering member.

Because a moving member may accumulate information for each area it visits, two options can be adopted depending on whether the provision of backward secrecy is necessary or not for controlling access to an area's past security information (which could be used for malicious purposes). This protocol will thus consider both options of members moving with and without backward secrecy.

In the first option, where provision of backward secrecy is not necessary, no update of keying material will occur (in particular an AKM of a visited area does not need to re-key its area key). On the other hand, the second option requires an AKM to re-key its area key when a group member moves in.

In dynamic mobile environments, group members may frequently move between a number of areas while still remaining in a group session. Every time a member moves into an area, re-keying of an area key may need to occur. As frequent re-keying may cause disruption of group communication, it may be necessary to keep track of the mobility of a highly dynamic group member. This can be useful to avoid frequent re-keying of an area key. This protocol will thus consider this circumstance.

To facilitate this, each key manager in a domain (DKM and AKMs) will need to maintain a list which contains information on the moving member, such as area(s) visited (which also indicates how many area keys the member possesses). This list is referred to as a *mobility list* (*MobList*) and is discussed in *Section 9.1.4*. In cases where a group member moves back into an area that it recently visited, an AKM can look up its *MobList*, and if the member is on the list (and is still a valid member of the multicast group) the AKM can determine that the member is a returning member who is moving back into the area. In this case, the area key of that particular area may not need to be re-keyed.

On the other hand, the area key of the visited area may need to be re-keyed if such a member is not on the *MobList*, and it is the member's first time entering the area.

Note that this process is completely separate to the re-keying that needs to occur whenever there is a change in group membership due to new member joins and/or existing member leaves. In any of these circumstances, group members (including the moving member) will need to be updated with new cryptographic keys.

The main functional requirements of this protocol are divided into the two *moving* options, as follows:

(a) For members moving *without* backward secrecy, the main functional requirements are to:

- transfer a group member from one area to another area.
- deliver an area key of a visited area to a moving member.

(b) For members moving *with* backward secrecy, in addition to those specified in (a), the other main functional requirements are to:

- initiate a re-keying of an area key of a visiting area.
- deliver a new area key of a visited area to a moving member.

A further protocol is required to govern the establishment of a short-term key to support host mobility.

For secure transfer of a group member from one area to another, security information (such as cryptographic keys) may need to be exchanged between communicating entities (in particular between a moving member and an AKM of a visited area) via a secure channel. This requires both entities (the moving member and the AKM of a visited area) to share a common secret key prior to the commencement of the move protocol. This type of key is referred to as a *session mobility* key (see *Section 8.3.6.2*).

This protocol governs the establishment of a session mobility key between a group member that wishes to move to another area and an AKM of a visited area. The generation and initial distribution of a session mobility key is conducted by a DKM in a domain and the key is delivered to the intended member via an AKM in the area where the member is currently residing. The same key is delivered to an AKM of a visited area by the DKM.

(c) The main functional requirements of this protocol are to:

  - establish a session mobility key between a moving member and an AKM of a visited area.
  - deliver a session mobility key to a moving member and to an AKM of a visited area.

The main security requirements of this protocol are to ensure that:

- only transfers from authorized group members are processed.
- communications between the group member and the area key manager are secure.
- communications between the area key manager and the domain key manager are secure.
- for member moving without backward secrecy, the distribution of area key A_Key of the visited area to the moving member is protected.
- for member moving with backward secrecy, the distribution of new area key A_Key_{new} of the visited area to the moving member is protected.
- the establishment of session mobility key between a moving member and an area key manager of the visited area is secure.
- the distribution of session mobility key to the moving member and to the area key manager of the visited area is protected.

8.4.5 Re-keying

This section describes re-keying protocols of the cryptographic keys that are needed for a multicast group communication, in particular the traffic key T_Key and the area key A_Key.

As mentioned in *Section 4.3.2*, re-keying may need to occur if the provision of backward and/or forward secrecy is necessary for a multicast group communication. In addition, dynamic mobile environments may also require a multicast group to re-key (or update) group members with a new set of keys (see *Section 5.3*).

We explicitly describe re-keying operations for the following two types of key:

(a) **Re-keying the traffic key**

This protocol governs re-keying of a traffic key of a multicast group. In particular, re-keying of a traffic key occurs to satisfy the design requirements of *Protocol II(b): Member joining with backward secrecy* (see *Section 9.4*), and *Protocol III(b): Member leaving with forward secrecy* (see *Section 9.6*).

Note that re-keying of a traffic key does not always need to occur during a host mobility protocol (see *Section 8.4.4*).

The main functional requirements of this protocol are to distribute a new traffic key to:

- all group members of a multicast group (including the newly joined member) during the commencement of *Protocol II(b)* (see *Section 9.4*).
- all remaining group members of a multicast group (excluding the leaving member) during the commencement of *Protocol III(b)* (see *Section 9.6*).

The main security requirements of this protocol are to ensure that:

- the distribution of the new traffic key T_Key_{new} to all group members of a multicast group (including the newly joined member) during *Protocol II(b)* (see *Section 9.4*) is protected.

- the distribution of the new traffic key T_Key_{new} to all remaining members (excluding the leaving member) during *Protocol III(b)* (see *Section 9.6*) is protected.

(b) **Re-keying the area key**

This protocol governs re-keying of an area key. In particular, re-keying of an area key occurs not only to satisfy the design requirements of *Protocol II(b)* (see *Section 9.4*) and *Protocol III(b)* (see *Section 9.6*), but also *Protocol IV(b): Member moving with backward secrecy* (see *Section 9.8*).

The main functional requirements of this protocol are to distribute a new area key to:

- all group members in an area (including the newly joined member) during the commencement of *Protocol II(b)* (see *Section 9.4*).
- all remaining group members in an area (excluding the leaving member) during the commencement of *Protocol III(b)* (see *Section 9.6*).
- all group members in a visited area (including the moving member) during the commencement of *Protocol IV(b)* (see *Section 9.8*).

The main security requirements of this protocol are to ensure that:

- the distribution of the new area key A_Key_{new} to all group members in an area (including the newly joined member) during *Protocol II(b)* (see *Section 9.4*) is protected.
- the distribution of the new area key A_Key_{new} to all remaining members in an area (excluding the leaving member) during *Protocol III(b)* (see *Section 9.6*) is protected.
- the distribution of the new area key A_Key_{new} to all group members in a visited area (including the moving member) during *Protocol IV(b)* (see *Section 9.8*) is protected.

As discussed in *Section 6.2.5*, re-keying operations in a domain can be divided into two levels, depending on the entity responsible for re-keying. In this protocol, re-keying is initiated by trusted key managers:

(a) *At the domain level*
A domain key manager (DKM) is responsible for the initiation and control of the re-keying of keys, in particular the traffic keys.

(b) *At the area level*
Each area key manager (AKM) is responsible for the initiation and control of the re-keying of an area key within its area.

Only re-keying of a traffic key and an area key are described here because these keys are the ones which need regular updates to achieve secure group communication. Other cryptographic keys such as *Domain-Area key, Domain key* and *Area-Member key* (which are long-term keys) are not explicitly treated here because they do not need to be updated so often.

8.5 Summary

In this chapter, we determined the scope of our GKMF, and looked at the main components in the proposed architecture. We have also stated our reasons for choosing the design and have described each of the underlying protocol requirements identified within our GKMF.

In the next chapter we specify the design of the main protocols.

Chapter 9

GKMF for WMobE: GKMF Protocols

This chapter specifies the design of the main protocols identified with our group key management framework (GKMF).

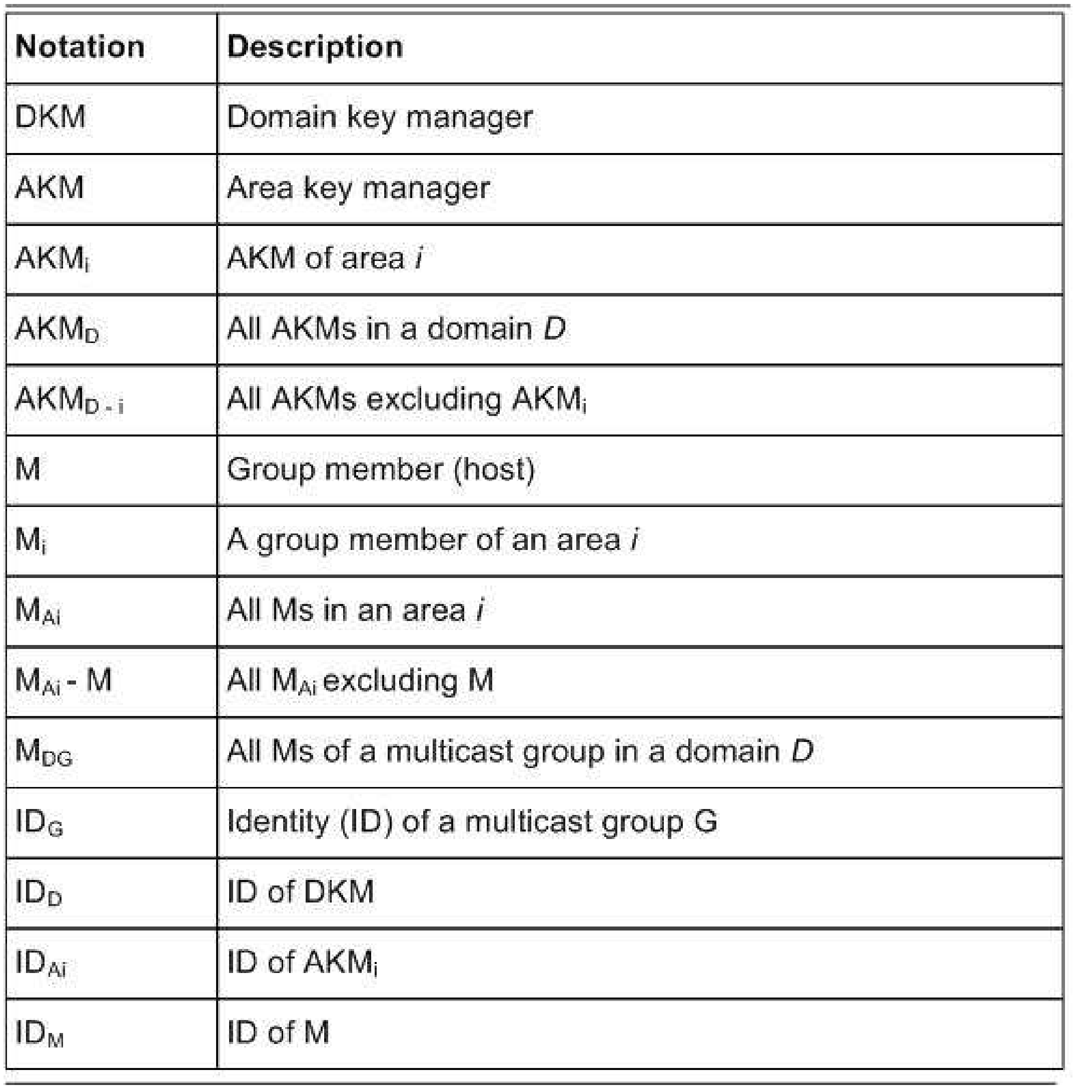
We organize this chapter as follows. In *Section 9.1* we introduce the notation and assumptions used in the protocol designs. Our protocol designs begin in *Section 9.2* with creation of new group and initial distribution of keys. In *Section 9.3* and *Section 9.4*, we specify the protocols for new member joining without and with provision of backward secrecy. In *Section 9.5* and *Section 9.6*, we describe the protocols for existing member leaving without and with provision of forward secrecy. In *Section 9.7* and *Section 9.8*, we specify the protocols for members moving without and with provision of backward secrecy. Finally, in *Section 9.10* and *Section 9.11*, we describe the re-keying protocols for traffic key and area key.

9.1 Introduction

In this section, we describe notation as well as important assumptions that we will make during our protocol designs. We also introduce the use of *lists* as part of our protocol design, in particular for members moving to other areas.

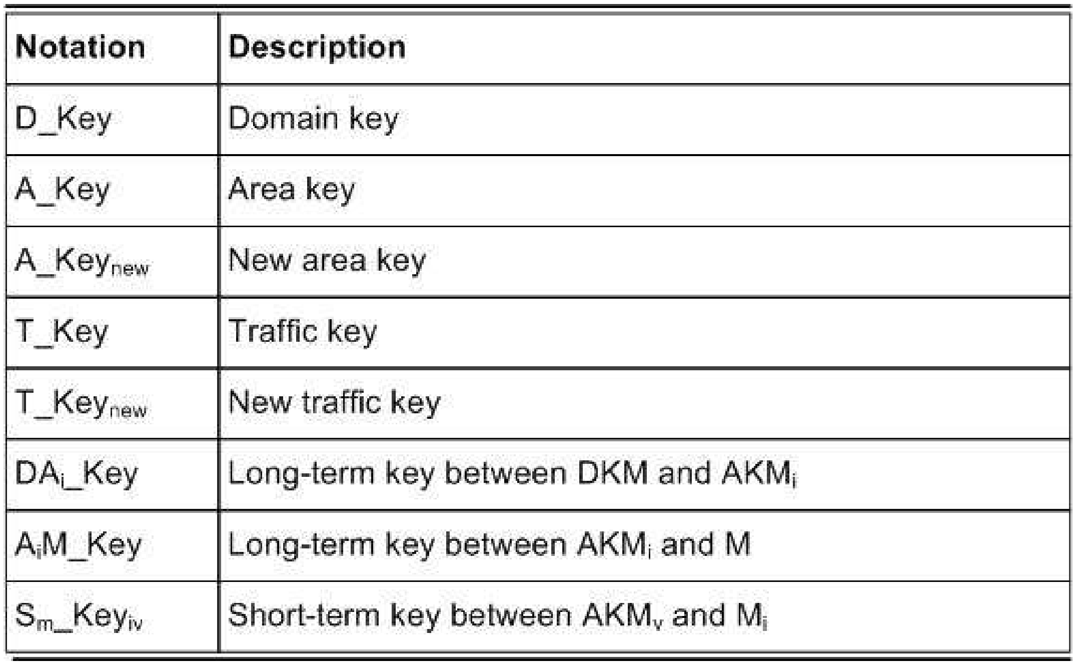
9.1.1 Notation

In this section, we introduce the notation that we will use for describing the proposed protocols. This is presented in the form of four tables. *Table 9.1* presents the notations for *entities*. *Table 9.2* gives the notations for *keys*. *Table 9.3* shows the notations for *other operatives* necessary for group operations. *Table 9.4* presents the notations for *tokens* that we use for distribution of security parameters during group operations.



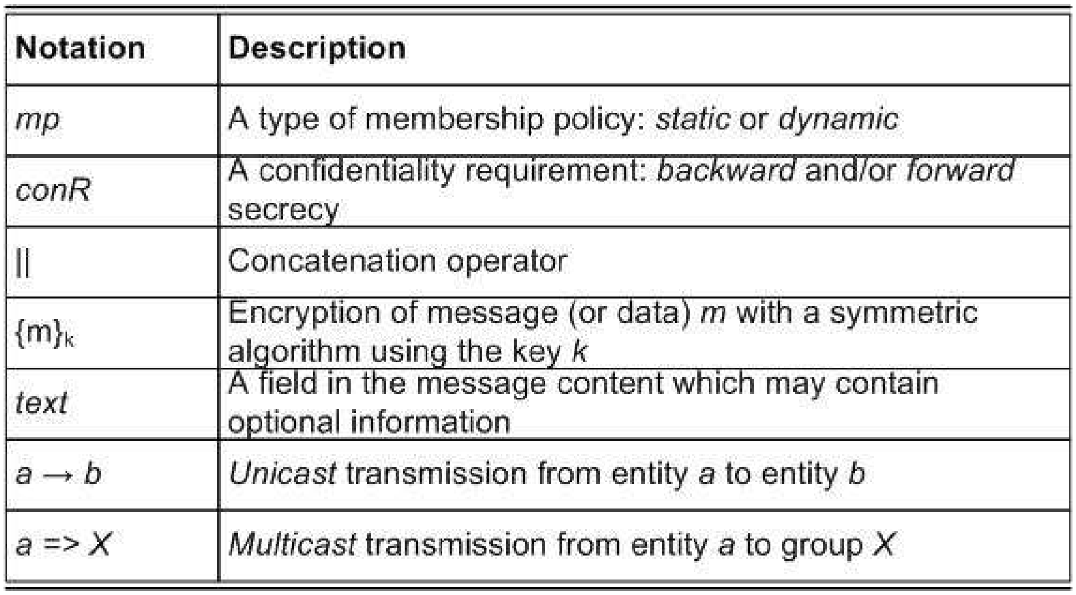
Notation	Description
DKM	Domain key manager
AKM	Area key manager
AKM_i	AKM of area *i*
AKM_D	All AKMs in a domain *D*
$AKM_{D - i}$	All AKMs excluding AKM_i
M	Group member (host)
M_i	A group member of an area *i*
M_{Ai}	All Ms in an area *i*
M_{Ai} - M	All M_{Ai} excluding M
M_{DG}	All Ms of a multicast group in a domain *D*
ID_G	Identity (ID) of a multicast group G
ID_D	ID of DKM
ID_{Ai}	ID of AKM_i
ID_M	ID of M

Table 9.1: Summary of notation for entities.



Notation	Description
D_Key	Domain key
A_Key	Area key
A_Key$_{new}$	New area key
T_Key	Traffic key
T_Key$_{new}$	New traffic key
DA$_i$_Key	Long-term key between DKM and AKM$_i$
A$_i$M_Key	Long-term key between AKM$_i$ and M
S$_m$_Key$_{iv}$	Short-term key between AKM$_v$ and M$_i$

Table 9.2: Summary of notation for keys.

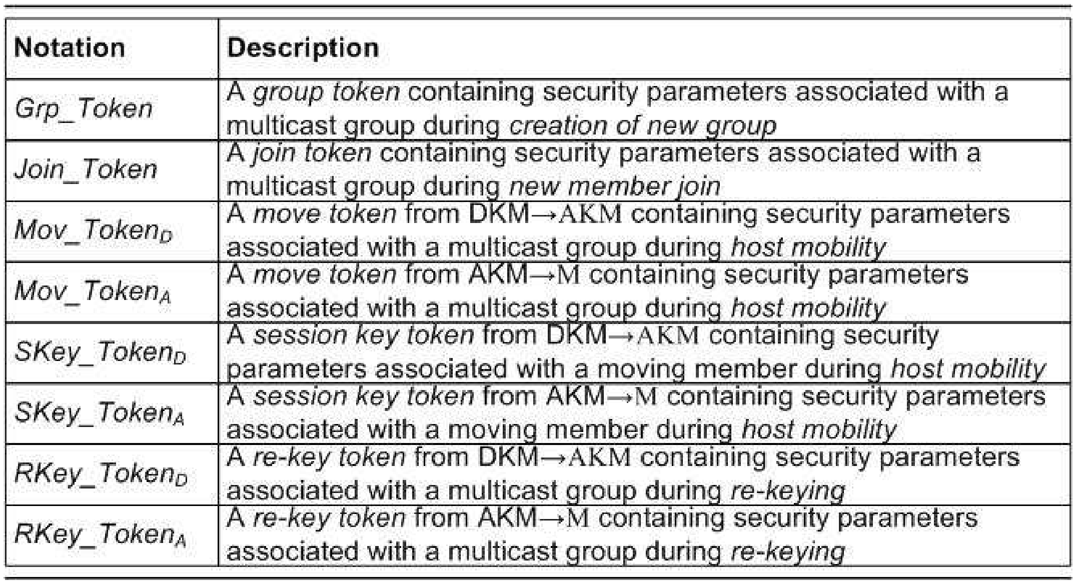


Notation	Description
mp	A type of membership policy: *static* or *dynamic*
conR	A confidentiality requirement: *backward* and/or *forward* secrecy
||	Concatenation operator
$\{m\}_k$	Encryption of message (or data) *m* with a symmetric algorithm using the key *k*
text	A field in the message content which may contain optional information
a → *b*	*Unicast* transmission from entity *a* to entity *b*
a => *X*	*Multicast* transmission from entity *a* to group *X*

Table 9.3: Summary of notation for other operatives.

9.1.2 Provision of Security Services

The security requirements identified within the framework in *Section 8.2* include the requirements for *entity authentication*, *backward and forward secrecy*,



Notation	Description
Grp_Token	A *group token* containing security parameters associated with a multicast group during *creation of new group*
Join_Token	A *join token* containing security parameters associated with a multicast group during *new member join*
Mov_Token_D	A *move token* from DKM→AKM containing security parameters associated with a multicast group during *host mobility*
Mov_Token_A	A *move token* from AKM→M containing security parameters associated with a multicast group during *host mobility*
$SKey\_Token_D$	A *session key token* from DKM→AKM containing security parameters associated with a moving member during *host mobility*
$SKey\_Token_A$	A *session key token* from AKM→M containing security parameters associated with a moving member during *host mobility*
$RKey\_Token_D$	A *re-key token* from DKM→AKM containing security parameters associated with a multicast group during *re-keying*
$RKey\_Token_A$	A *re-key token* from AKM→M containing security parameters associated with a multicast group during *re-keying*

Table 9.4: Summary of notation for tokens.

as well as *data integrity* security services.

As discussed in *Section 4.5*, we describe how these security requirements can be achieved via the use of various security mechanisms such as those in ISO 7498-2 (ISO, 1989), ISO/IEC 11770-1 (ISO, 1996a), ISO/IEC 11770-2 (ISO, 1996b), BS ISO/IEC 9798-1 (BS, 1997), ISO/IEC 9798-2 (ISO, 1999a) and ISO/IEC 9798-4 (ISO, 1999b).

The provision of the aforementioned security services based on symmetric key cryptography is as follows:

- **Entity authentication**
  Entity authentication can be achieved when an entity to be authenticated corroborates its claimed identity by demonstrating its knowledge of a secret key. The mechanisms that can be used to provide this include ISO/IEC 9798-2 (ISO, 1999a), ISO/IEC 9798-4 (ISO, 1999b):

  (a) *Encryption (and decryption)*
  By careful application of encryption mechanisms on specific data another entity sharing the secret key can corroborate its claimed identity.

(b) *Message authentication codes (MACs)*

By applying a key to specific data to obtain a MAC value, this value can be compared to the value sent by the other entity, who will corroborate the other's identity when the value received is the same as the value generated.

Note that demonstration of use of a key does not guarantee the message is not a replay from previous sessions. Thus, the cryptographic computations must also include *time variant parameters* as those in ISO/IEC 9797-2 (ISO, 1994a), ISO/IEC 9797-1 (ISO, 1994b), ISO/IEC 11770-2 (ISO, 1996b), ISO/IEC 9798-2 (ISO, 1999a) and ISO/IEC 9798-4 (ISO, 1999b), such as time stamps, sequence numbers or, random numbers. For example, if a received message has a time stamp that falls outside an agreed *window of acceptance* (BS, 2002), then the message received is considered as not fresh and is discarded from further processing.

Entity authentication is typically required at two levels:

(a) *Domain level*

At the domain level, this involves all key managers (the DKM and all AKMs). Since all key managers in the domain are assumed to be trusted by the framework, it is essential for all key managers to be certain of each other's identity. Both entities (the DKM and every AKM in the domain) may verify each other's identity without the involvement of third party, such as by adopting ISO/IEC 9798-2 (ISO, 1999a).

These mechanisms require both DKM and each AKM to share a long-term common secret key prior to the authentication mechanism taking place (see *Section 8.3.6.1*, for long-term keys). This will occur during the assignment of AKM as a key manager in the domain, before the creation of any multicast groups occurs.

(b) *Area level*

At the area level, any host wishing to join a multicast group will first have to establish its identity with an AKM. As in ISO/IEC 9798-2 (ISO, 1999a), this can be achieved either as follows:

- *With the involvement of a third party.* In this case, entity authentication can be established between the host and AKM with the assistance of the DKM. For this to work, both host and AKM must each share a long-term common secret key with the DKM prior to invoking the authentication mechanism (see *Section 8.3.6.1*, for long-term keys). Through the DKM, a common secret key will then be established, to be used between the host and the AKM.

  - *Without the involvement of a third party.* Without the involvement of the DKM, this mechanism requires both host and AKM to share a long-term common secret key prior to the authentication mechanism taking place (see *Section 8.3.6.1*, for long-term keys). This mechanism is similar to the one that occurs between the key managers at the domain level.

- **Backward and forward secrecy**
  The provision of backward and forward secrecy is supported by *confidentiality* services, where it can be achieved by careful application of encryption mechanisms as in ISO 7498-2 (ISO, 1989). The provision of backward and/or forward secrecy is primarily determined by suitable key management techniques for updating (or, re-keying) with new keys whenever there is a change in group membership.

- **Data origin authentication**
  The provision of data origin authentication (and hence data integrity) can be achieved by using standard shared key authentication mechanisms such as MACs in ISO 7498-2 (ISO, 1989), ISO/IEC 9797-2 (ISO, 1994a) and ISO/IEC 9797-1 (ISO, 1994b).

  In our proposal, by using such mechanisms we are particularly concerned with two types of data authentication, as follows (see also *Section 4.5*):

  (a) *Source and data authentication*
  This type of authentication enables a group member to verify the source of the message received and that the message was not modified during transit by anyone, including other group members.

  Thus, if data origin authentication is deemed necessary, the best mechanism that can be used to provide this is a MAC, because only the entities who hold the secret key can compute the same MAC value. Using such mechanisms, an AKM can be sure that the cryptographic keys it receives originated from the DKM that they claim to be from.

(b) *Group authentication*

Group authentication enables a group member to verify that the data or message he receives originated from some member in the group, and that it was not modified by anyone that is not a group member.

This type of authentication may be adequate if there are circumstances when it is not essential to provide the precise origin of data one entity receives. For example, during actual data communication between the group members, where no security information is being exchanged.

For these mechanisms to work, all group members are required to share a common secret key prior to implementing the group communication.

In the event that either of these mechanisms fail, the recipient can conclude that the message received has been tampered with, and that the integrity of the message can no longer be guaranteed.

9.1.3 Focus of Simplified Protocol Descriptions

Of all the previously discussed security services, we are primarily interested in the confidentiality services, in particular the provision of backward and/or forward secrecy.

Our protocol descriptions will be simplified in order to highlight how the keys in the GKMF are utilized to provide confidentiality services. This is primarily for reason of clarity of key usage within the GKMF.

We do not explicitly demonstrate the provision of other security services such as entity authentication and data integrity, because these can be incorporated into the protocols using generic techniques that are not specific to group communication. For example:

(a) The provision of data origin authentication can easily be provided in all our protocols by adding a MAC to each message, as discussed in *Section 9.1.2*. MAC keys can easily be derived from symmetric keys in use.

(b) The provision of entity authentication (when necessary) can then be incorporated into these protocols by careful implementation of time variant parameters into the MAC value, and following standards such as those identified in *Section 9.1.2*.

Thus, the protocols in this chapter should be regarded as a blueprint for full protocols which can be derived using our skeleton protocol specifications.

9.1.4 List(s) Management

In this section, we introduce an important concept that we will use as part of our group key management protocol designs; *lists*. We describe the management of lists, and why they are useful, in particular within the existing member leaving protocol and the host mobility protocol.

We propose two types of *list*, each of which is described as follows:

(a) **HisList**
HisList (*history list*) is maintained by a domain key manager (DKM) and contains information of members who have left a multicast group. Each time a member leaves a multicast group, the following information is logged in *HisList*:

- ID of the group member,
- ID of the multicast group the member is leaving,
- ID of the area that the member leaves from,

- Type of leave; either *voluntary* or *involuntary* (see *Section 4.3.2*).

The information in *HisList* is used by key managers in a domain for future reference, in order to keep track of the group members who leave, as well as the reasons for leaving a multicast group. History lists are particularly useful when members try to re-join a group that they previously left.

In this case, a key manager may use its *HisList* to determine the reason why a user previously left the group. In the case of involuntary leaves the member may be refused permission to re-join the group.

Examples of use of *HisList* can be seen in *Section 9.5* and *Section 9.6*.

(b) **MobList**

MobList (*mobility list*) is maintained by key managers (DKM and AKMs) in a domain and contains information on group members that move from one area to another (while still remaining in group sessions). Each time a member moves from one area to the next, the following information is logged in *MobList*:

- ID of the moving member,
- ID of the multicast group joined by the member,
- ID of the area that a member is *moving from* (which corresponds to an AKM of that area),
- ID of the visited area that a member is *moving to*.

The information in *MobList* is used by key managers (DKM and AKMs) to efficiently manage members that may frequently move between a number of areas while still remaining in a group session. As re-keying of an area key may need to occur every time a member moves into an area, and frequent re-keying may cause disruption in group communication, *MobList* can be used to keep track of host mobility and frequent re-keying can be avoided every time a member moves back into an area that it recently visited. This is because when the same member moves back into that area, an AKM of a visited area can determine (by looking up its *MobList*) whether the member is a returning member who is just moving back into the area, in which case re-keying of the area's key may not need to take place.

Examples of use of *MobList* can be seen in *Section 9.7* and *Section 9.8*.

We assume that these lists are kept and maintained by key managers in secure environments.

9.1.5 Use of Text Field(s)

In this section we describe the use of *text* field(s) specified in the protocol messages of the protocol designs (see *Section 9.2* onwards).

Text fields may contain other information that is also needed in the protocol design but is not explicitly shown in the protocol because the content is dependent upon specific applications. This type of information is considered optional, and does not affect the overall design of the protocols.

Information that a *text* field may contain includes those specified in ISO/IEC 11770-1 (ISO, 1996a), ISO/IEC 11770-2 (ISO, 1996b), BS ISO/IEC 9798-1 (BS, 1997), ISO/IEC 9798-2 (ISO, 1999a) and ISO/IEC 9798-4 (ISO, 1999b), such as:

- A *key lifetime*; indicating the validity period of a key,
- A *key identifier*; indicating the key usage,
- A *cryptographic method*; indicating the type of algorithm used,
- A *check value*; such as a MAC value to check the integrity of a message,
- A *time variant parameter*; such as a time stamp indicating a message's age, which can be used to verify that a message is not a replay.

9.2 Protocol I: Creation of New Group and Initial Distribution of Keys

This protocol describes the creation of a multicast group by a host M, and initial distribution of a traffic key T_Key and an area key A_Key by the key

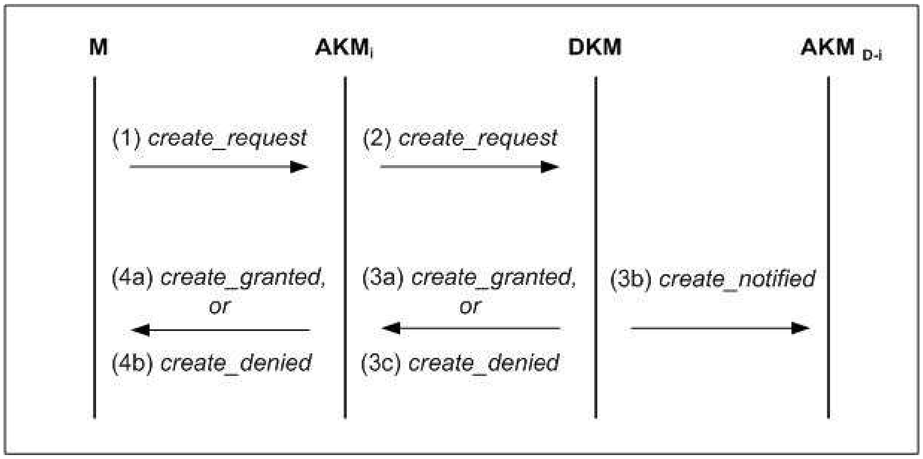


Figure 9.1: Creation of New Group and Initial Distribution of Keys protocol message flow.

manager to M. Host M is considered as the first member of the multicast group.

The message flow of this protocol is depicted in *Figure 9.1*, and steps involved are described as follows:

Step 1: A host M that wishes to form a multicast group sends a $create\_request$ message to the area key manager AKM$_i$ encrypted under Area-Member key A_iM_Key (which has been established earlier, see *Section 8.3.6.1*):

$$M \rightarrow AKM_i : ID_M \| \{ID_{A_i} \| ID_M \| text\} A_iM\_Key.$$

Step 2: On receipt, AKM$_i$ checks the message by decrypting it using the secret key A_iM_Key shared with the host M. AKM$_i$ then passes the $create\_request$ message to DKM encrypted under Domain-Area Key DA_i_Key (which has been established earlier, see *Section 8.3.6.1*), along with the ID of the requesting entity:

$$AKM_i \rightarrow DKM : ID_{A_i} \| \{ID_{A_i} \| ID_M \| text\} DA_i\_Key.$$

Step 3: On receipt, DKM performs the following:

(a) DKM looks up the list of AKM$_D$ with their corresponding secret keys, and checks the message by decrypting it using the secret key

DA_i_Key shared with AKM$_i$. Assuming that DKM keeps a valid list of potential hosts, and assuming that host M is in the list, M is allowed to form a multicast group.

DKM then sends a *create_granted* message to AKM$_i$ containing the information of the newly created multicast group, including the traffic key T_Key, as well as the group policy in the form of Grp_Token:

$$DKM \rightarrow AKM_i : ID_D \| \{Grp\_Token \| ID_D \| text\} DA_i\_Key,$$

where $Grp\_Token = \{ID_G \| ID_{A_i} \| ID_M \| T\_Key \| mp \| conR \| text\}$.

(b) DKM then notifies other AKMs of the creation of the multicast group via the $create\_notified$ message, along with the Grp_Token of the multicast group. DKM can send this message via two ways:

- *unicast*, where DKM sends the message to each AKM individually (encrypted under *Domain-Area* key) as in *Step 3(a)*.
- *multicast*, where DKM sends a single message to all AKMs (encrypted under the domain key D_Key).

$$DKM \Rightarrow AKM_D : ID_D \| \{Grp\_Token \| ID_D \| text\} D\_Key.$$

Otherwise, DKM sends a *request_denied* message to AKM$_i$.

Note that the *create_granted* and $create\_notified$ messages (see *Figure 9.1*) can be sent altogether by DKM to all AKMs (including AKM$_i$) via a *multicast* message (as in *Step 3(b)*).

Step 4: On receipt, AKM$_i$ performs the following:

(a) AKM$_i$ checks the message by decrypting it with the secret key DA_i_Key which it shares with DKM, and assuming that the *create_request* is granted, AKM$_i$ obtains the Grp_Token of the new multicast group.

(b) AKM$_i$ then generates an area key A_Key and updates the Grp_Token with the area key as follows:

$$Grp\_Token = \{ID_G \| ID_{A_i} \| ID_M \| T\_Key \| A\_Key \| mp \| secR \| text\}.$$

AKM$_i$ then sends it via a *create_granted* message to M encrypted under A_iM_Key:

$$AKM_i \rightarrow M : ID_{A_i} \| \{Grp\_Token \| text\} A_iM\_Key.$$

Otherwise, AKM$_i$ sends a *create_denied* message to M.

Step 5: On receipt, M checks the message by decrypting it with A_iM_Key. Assuming that the request to create a multicast group is granted, M obtains the keys along with other information via the Grp_Token.

In summary, assuming that the *join* request is successful, at this point a new multicast group is created, with DKM as the domain key manager, AKM$_i$ as the area key manager, and M as the group initiator. The keys that are distributed through the protocol are the traffic key T_Key and the area key A_Key.

9.3 Protocol II(a): New Member Joining without Backward Secrecy

This protocol describes a new join of a host to a multicast group with no consideration to secure access to the previous data traffic, in other words no provision of backward secrecy. The protocol also includes the delivery of a traffic key T_Key and an area key A_Key to the newly joined group member M.

Throughout this protocol, we make the following assumption:

- Any request to join the group only occurs after the successful creation of the multicast group using *Protocol I* (*Section 9.2*).

The message flow of this protocol is depicted in *Figure 9.2*, and the steps involved are described as follows:

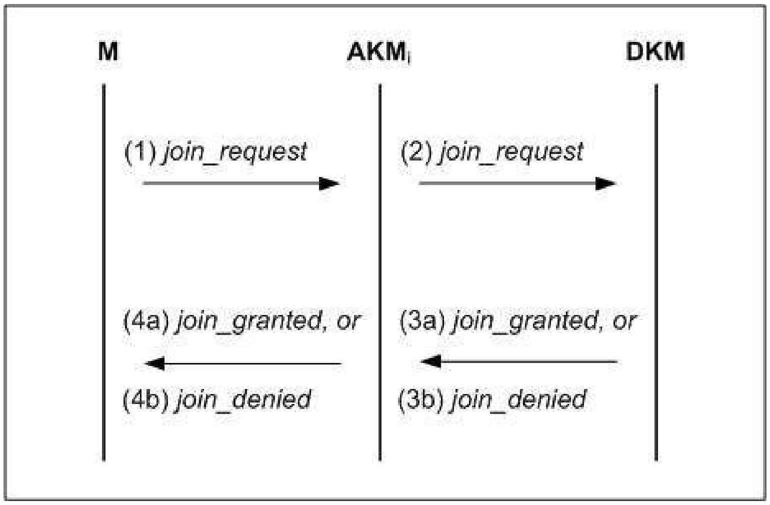


Figure 9.2: Member Joining without Backward Secrecy protocol message flow.

Step 1: A host M that wishes to join a multicast group sends a *join_request* message to the area key manager AKM$_i$ encrypted under A_iM_Key (see *Section 8.3.6.1*):

$$M \rightarrow AKM_i : ID_M \| \{ID_G \| ID_{A_i} \| ID_M \| text\} A_iM\_Key.$$

Step 2: On receipt, AKM$_i$ performs the following:

(a) AKM$_i$ checks the message by decrypting it with the secret key A_iM_Key shared with M. AKM$_i$ then passes the *join_request* message to DKM encrypted under Domain-Area key DA_i_Key (see *Section 8.3.6.1*), along with the ID of the requesting entity, as well as the ID of the multicast group for joining:

$$AKM_i \rightarrow DKM : ID_{A_i} \| \{ID_G \| ID_{A_i} \| ID_M \| text\} DA_i\_Key.$$

(b) Assuming that host M is permitted to join the group, AKM$_i$ sends a *join_granted* message to M encrypted under A_iM_Key, along with the current keys (the traffic key T_Key and the area key A_Key) in the form of $Join\_Token$:

$$AKM_i \rightarrow M : ID_{A_i} \| \{Join\_Token \| text\} A_iM\_Key,$$

where $Join\_Token = \{ID_G \| ID_{A_i} \| ID_M \| T\_Key \| A\_Key \| text\}$. Otherwise, AKM$_i$ sends a *join_denied* message to M.

Note that the keys that are currently in use by group members are delivered to the newly joined member.

Step 3: Upon receiving the *join_request* message, DKM checks the message by decrypting it with the secret key DA_i_Key which it shares with AKM$_i$, and assuming that host M is granted permission to join the multicast group, DKM sends a *join_granted* message to AKM$_i$.

Note that it is not necessary for other AKMs to be notified of the new join as it does not affect the overall group operation in the domain. New joins with no provision of backward secrecy simply require DKM to add a new member to a particular multicast group.

Step 4: Upon receiving the message from AKM$_i$, M checks the message by decrypting it with his secret key A_iM_Key to obtain the $Join\_Token$ which, in particular, contains the cryptographic keys needed for the group communication.

In summary, assuming that host M is granted permission, using this protocol M joins a multicast group within an area where AKM$_i$ is the area key manager and DKM is the domain key manager. Group member M is also given the current set of cryptographic keys; the traffic key T_Key and the area key A_Key.

9.4 Protocol II(b): New Member Joining with Backward Secrecy

This protocol describes a new join of a host to a multicast group with consideration to preventing access to the previous data traffic from the newly joined member, in other words provision of backward secrecy. This protocol also includes the delivery of a new traffic key T_Key_{new} and a new area key A_Key_{new} to the newly joined member M and other group members (if any) in the area where the join occurs, as well as across the domain.

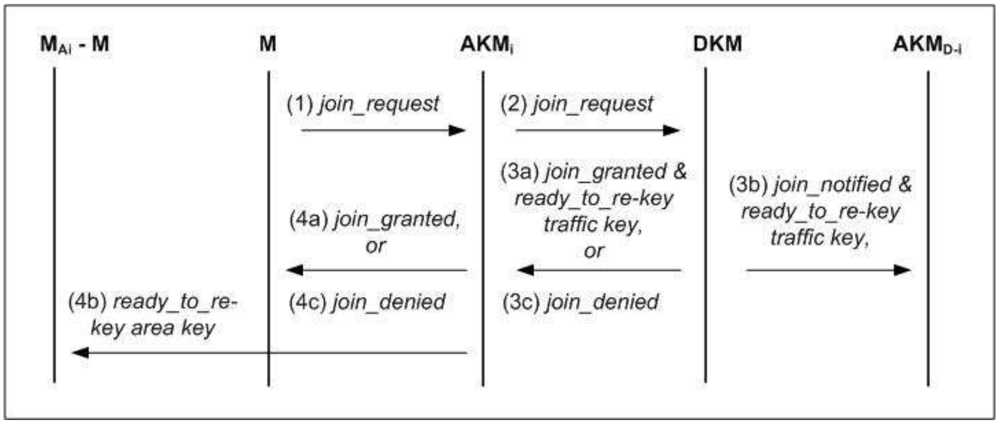

M_Ai - M
M
AKMi
DKM
AKMD-i
(1) join_request
(2) join_request
(3a) join_granted & ready_to_re-key traffic key, or
(3b) join_notified & ready_to_re-key traffic key,
(4a) join_granted, or
(4c) join_denied
(3c) join_denied
(4b) ready_to_re-key area key

Figure 9.3: Member Joining with Backward Secrecy protocol message flow.

The message flow of this protocol is depicted in *Figure 9.3*. Due to similarity with *Protocol II(a)* (see *Section 9.3*), we only describe the differences as follows:

(a) In addition to *Step2* in *Protocol II(a)* (see *Section 9.3*), and assuming that the *join* is granted, AKMi_i must re-key its area key. To do so, it initiates *Protocol VI: Re-keying the area key* (see *Section 9.11*). This results in all existing members in that particular area obtaining the new area key A_Key_{new}.

Note that only the area where the new host joins the group is re-keyed with a new area key. (Since the change in current group membership occurs only within a particular area, other areas should not be affected.)

(b) AKM$_i$ then delivers the new area key A_Key_{new} in the $join\_granted$ message to M in the form of $Join\_Token$, as in *Step 2(b) Protocol II(a)* (see *Section 9.3*).

(c) In addition to *Step3*, upon receiving the $join\_request$ message from AKM$_i$, and assuming host M is allowed to join the multicast group, DKM initiates *Protocol V: Re-keying the traffic key* (see *Section 9.10*) and sends the $join\_granted$ message along with $ready\_to\_re-key$ traffic key to all AKMs in the domain (including AKM$_i$). This results in all AKMs and all group members in the domain obtaining a new traffic key T_Key_{new}.

DKM can send this message via two ways:

- *unicast*, where DKM sends the message to each AKM individually (encrypted under the *Domain-Area* key).
- *multicast*, where DKM sends a single message to all AKMs encrypted under the domain key D_Key.

(d) Upon receiving the new traffic key from DKM, AKM$_i$ sends the key to the newly joined member M.

Note that AKM$_i$ can delay the sending of the new area key A_Key_{new} to M until it receives the new traffic key T_Key_{new} from the DKM. In this case, AKM$_i$ can send both keys in the $Join\_Token$, such that $Join\_Token = \{ID_G \| ID_{A_i} \| ID_M \| A\_Key_{new} \| T\_Key_{new} \| text\}$.

The re-keying approach used in this protocol is a strict re-keying policy, where the re-keying of traffic key T_Key and area key A_Key are conducted immediately when the new joins occur. Taking a more relaxed approach, the re-keying tasks could be postponed until the next periodic re-keying (Decleene et al., 2001), (Zhang et al., 2002).

In summary, assuming that host M is granted permission, using this protocol M is joined to a multicast group within an area where AKM$_i$ is the area key manager and DKM is the domain key manager. Due to the provision of backward secrecy, DKM and AKM$_i$ must initiate the re-keying protocols of the current keys (traffic key and area key), which results in all group members in the domain obtaining a new traffic key, and all group members of an area where the join occurs obtaining a new area key. The new group member M is also given a current set of cryptographic keys: new traffic key T_Key_{new} and new area key A_Key_{new}.

9.5 Protocol III(a): Existing Member Leaving without Forward Secrecy

This protocol describes existing group members leaving the multicast group, with no consideration to secure access to future data traffic. Since no protection of future data traffic from the leaving members is necessary (no forward secrecy), members leaving is a relatively simple process, and no change in keying material occurs.

In this protocol, we make the following assumptions:

- It is assumed that there already exists an established multicast group.
- *Host mobility* may have occurred, and a group member may leave from a visited area (and not necessarily its local area).

The message flow for each type of leave discussed in *Section 8.4.3* is depicted in *Figure 9.4*, and the steps involved are described as follows:.

Step 1: From *Figure 9.4(a)*, a host M wishing to leave a multicast group sends a $leave\_notify$ message to the area key manager AKM$_i$ encrypted under Area-Member key A_iM_Key (see *Section 8.3.6.1*):

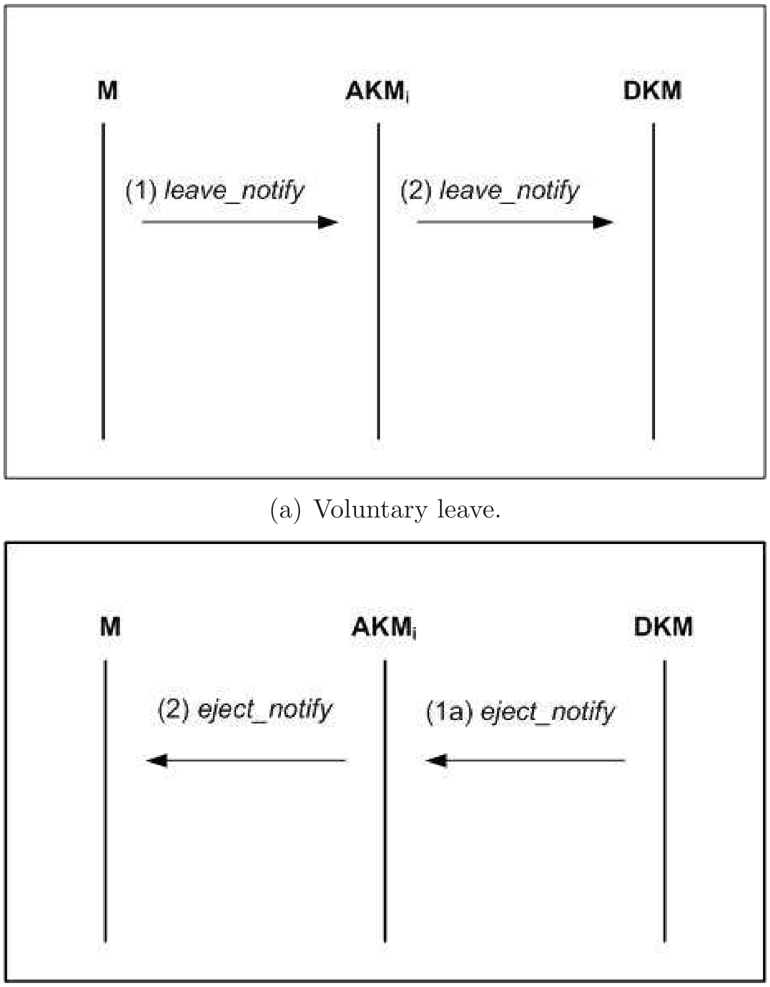
$$M \rightarrow AKM_i : ID_M \| \{ID_G \| ID_{A_i} \| ID_M \| text\} A_iM\_Key.$$

Step 2: On receipt, AKM$_i$ checks the message by decrypting it using the secret key A_iM_Key shared with the host M, and then passes the $leave\_notify$ message to the DKM encrypted under Domain-Area Key DA_i_Key (see *Section 8.3.6.1*), along with the ID of the group member and the ID of the multicast group from which M is leaving:

$$AKM_i \rightarrow DKM : ID_{A_i} \| \{ID_G \| ID_{A_i} \| ID_M \| text\} DA_i\_Key.$$

Step 3: On receipt, DKM checks the message by decrypting it using DA_i_Key, (which is shared with AKM$_i$), to obtain the information of the leaving member M.

Note that like hosts joining, it is not necessary for other AKMs to be notified of the leaving member. A member leaving with no provision for forward secrecy simply requires DKM to remove the member from a particular multicast group.



(a) Voluntary leave.

(b) Involuntary leave.

Figure 9.4: Member Leaving without Forward Secrecy protocol message flow.

In the case of involuntary leaves, DKM will initiate the process to eject a member by sending an $eject\_notify$ message along with the ID of the ejected member to AKM (where the ejecting member resides). The message is sent via a secure channel (see *Section 8.3.6.4*) protected under a *Domain-Area* key. Upon receiving the $eject\_notify$ message from DKM, the AKM notifies the particular member via an $eject\_notify$ message, protected under an *Area-Member* key (see *Figure 9.4(b)*).

For both cases (voluntary or involuntary leave), DKM will update its *HisList* with new information (such as the reason of leaving and/or the reason for ejecting) concerning the leaving member.

In summary, using this protocol, M leaves a multicast group. In case of voluntary leave, M initiates the protocol. Otherwise, it is initiated by the DKM. For both cases, information regarding the leaving member is logged into DKM's *HisList*.

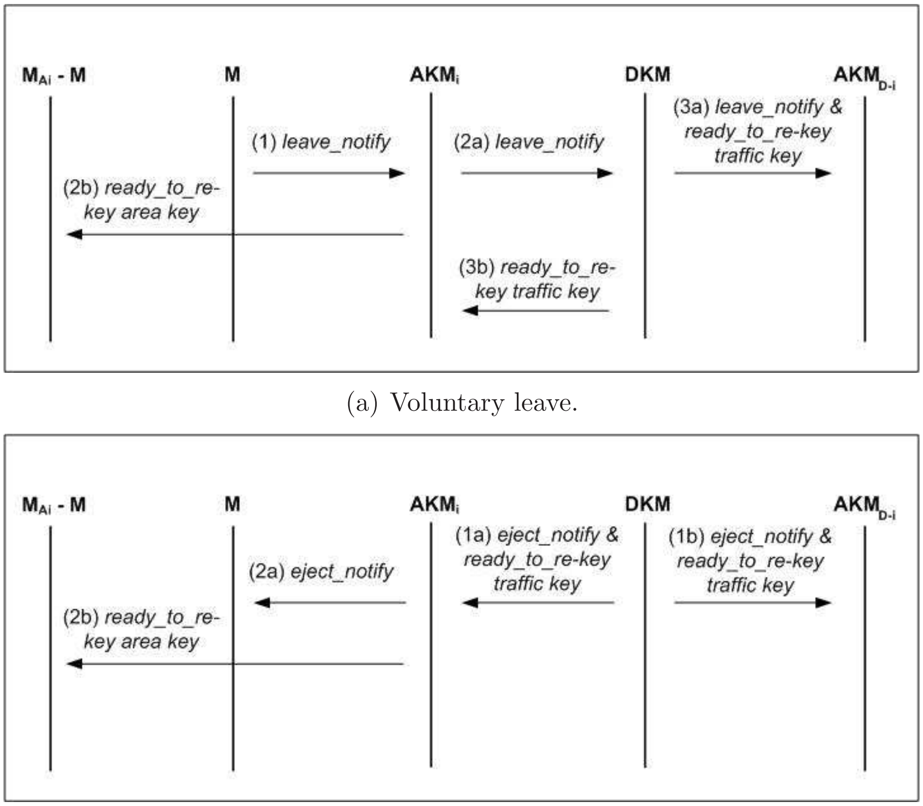
9.6 Protocol III(b): Existing Member Leaving with Forward Secrecy

This protocol describes group members leaving with consideration to controlling access to the future data traffic, in which case forward secrecy is necessary. A multicast group thus needs to be re-keyed with new keying material whenever an existing member leaves the group.

The assumptions made in *Protocol III(a)* (see *Section 9.5*) also hold for this protocol. The message flow of this protocol is depicted in *Figure 9.5*. Due to similarity with *Protocol III(a)* (*Section 9.5*), we just describe the differences as follows:

(a) In addition to *Step2* in *Protocol III(a)*, AKM$_i$ must re-key its area key A_Key, and for that it initiates *Protocol VI: Re-keying the area key* (see *Section 9.11*). This results in all remaining members (excluding the leaving member) in that particular area and in area(s) visited by the leaving member obtaining the new area key A_Key_{new}.

(b) In addition to *Step3*, upon receiving the $leave\_notify$ message from AKM$_i$, DKM initiates *Protocol V: Re-keying the traffic key* (see *Section 9.10*), which results in all AKMs and group members in the domain obtaining a new traffic key T_Key_{new}. As in (c) *Protocol II(b)* (see *Section 9.4*), DKM can send this message via two ways either by *unicast*, where the message is sent to each AKM individually encrypted under the *Domain-Area* key or, by *multicast*, where a single message is sent to all AKMs encrypted under the domain key D_Key.


MAi - M
M
AKMi
DKM
AKMD-i
(1) leave_notify
(2a) leave_notify
(3a) leave_notify & ready_to_re-key traffic key
(2b) ready_to_re-key area key
(3b) ready_to_re-key traffic key
MAi - M
M
AKMi
DKM
AKMD-i
(1a) eject_notify & ready_to_re-key traffic key
(1b) eject_notify & ready_to_re-key traffic key
(2a) eject_notify
(2b) ready_to_re-key area key

(a) Voluntary leave.

(b) Involuntary leave.

Figure 9.5: Member Leaving with Forward Secrecy protocol message flow.

As in *Protocol III(a)*, in the case of involuntary leaves, an $eject\_notify$ message along with the ID of the ejected member will be sent by DKM to all AKMs via secure channels (see *Section 8.3.6.4*). The area key manager (where the ejecting member resides) will then send an $eject\_notify$ message to member M (see *Figure 9.5(b)*). Similarly, DKM will update its *HisList* with the new information of the leaving member.

In summary, using this protocol, M leaves a multicast group. Like *Protocol III(a)*, in case of voluntary leave, M initiates the protocol. Otherwise, it is initiated by the DKM. For both cases, a new area key A_Key_{new} is obtained by the remaining members in the area (where the leave occurs and area(s) visited by the leaving member), and a new traffic key T_Key_{new} is obtained by all AKMs and group members across a domain. For both cases, information regarding the leaving member is logged into DKM's *HisList*.

9.7 Protocol IV(a): Member Moving without Backward Secrecy

This protocol describes transfer of a group member from one area to another with no consideration to secure access to previous keys and group data traffic (in other words, no provision of backward secrecy). The protocol also includes the delivery of an area key of a visited area to the moving member.

Throughout this protocol, we make the following assumptions:

- It is assumed that there already exists an established multicast group, and that a member may be in its local area, or in its visited area at the time of the move.
- To distinguish between the local and visited areas and the areas involved in host mobility, we use the following terminology:
  - (i) The area where a group member is *moving from* is referred to as the *leaving area*.
  - (ii) The area that a group member is *moving into* is referred to as the *joining area*.
- The generic notation that we use to differentiate the *entities* from both affected areas is as follows:
  - (i) *For the leaving area:*
    An area key manager and a group member are referred to as AKM$_i$ and M_i (with ID_{M_i} is the ID of member M_i), and their associated keys are an area key A_Key_i and a unique Area-Member key $A_iM_i\_Key$.
  - (ii) *For the joining area:*
    An area key manager is referred to as AKM$_v$, and is associated with an area key A_Key_v (and a new area key A_Key_{vnew}). Other group members in the area are referred to as M_{Av}.

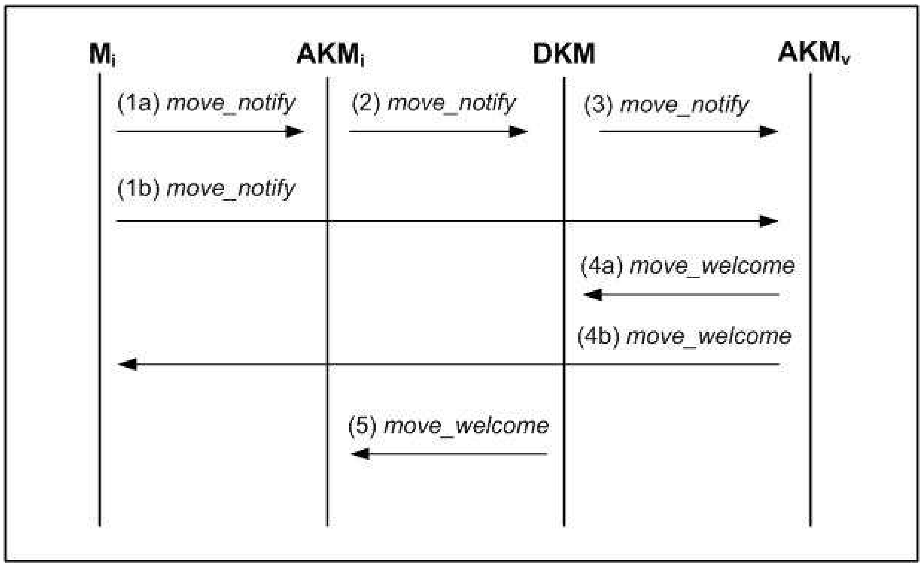


Figure 9.6: Member Moving without Backward Secrecy protocol message flow.

- It is assumed that a moving member M_i and an AKM$_v$ have securely established a shared short-term session mobility key $S_{m-}Key_{iv}$ prior to moving (see *Section 9.9*).

The message flow of this protocol is depicted in *Figure 9.6*, and the steps involved are described as follows:

Step 1: A group member M_i that wishes to move into another area sends a $move\_notify$ message along with the ID of the area that he is moving into (which corresponds to the ID_{A_v}) to:

(a) Its current area key manager AKM$_i$ protected under an Area-Member key $A_iM_{i-}Key$:

$$M_i \rightarrow AKM_i : ID_{M_i} \| \{ID_G \| ID_{A_i} \| ID_{A_v} \| ID_{M_i} \| text\} A_iM_{i-}Key.$$

(b) The area key manager of the visited area AKM$_v$ protected under a session mobility key $S_{m-}Key_{iv}$:

$$M_i \rightarrow AKM_v : ID_{M_i} \| \{ID_G \| ID_{A_i} \| ID_{A_v} \| ID_{M_i} \| text\} S_{m-}Key_{iv}.$$

Step 2: Upon receiving the *move_notify* message from M_i, AKM$_i$ checks the message by decrypting it with $A_iM_i\_Key$ and passes the message to DKM protected under Domain-Area key DA_i_Key (see *Section 8.3.6.1*):

$$AKM_i \rightarrow DKM : ID_{A_i} \| \{ID_G \| ID_{A_i} \| ID_{A_v} \| ID_{M_i} \| text\} DA_i\_Key.$$

Step 3: Upon receiving the message from AKM$_i$, DKM checks the message by decrypting it with DA_i_Key, and sends the *move_notify* message to AKM$_v$ along with the ID of M_i in the form of Mov_Token, where $Mov\_Token_D = \{ID_G \| ID_{A_i} \| ID_{A_v} \| ID_{M_i} \| text\}$ protected under the Domain-Area key DA_v_Key it shares with AKM$_v$:

$$DKM \rightarrow AKM_v : ID_D \| \{Mov\_Token_D \| ID_D \| text\} DA_v\_Key.$$

Step 4: Upon receiving the *move_notify* message from DKM and M_i, AKM$_v$ does the following:

(a) It checks the message from DKM by decrypting it with the Domain-Area key DA_v_Key it shares with the DKM.

(b) It checks the message from M_i by decrypting it with the session mobility key $S_m\_Key_{iv}$ it shares with M_i.

(c) Assuming that the checking is valid, AKM$_v$ looks up its MobList$_v$ and if M_i is not in the list (meaning that this is M_i's first time to enter the area), AKM$_v$ sends a *move_welcome* message to M_i and DKM as follows:

- To M_i, protected under the session key $S_m\_Key_{iv}$ along with its current area key A_Key_v in the form of Mov_Token:

  $$AKM_v \rightarrow M_i : ID_{A_v} \| \{Mov\_Token_A \| text\} S_m\_Key_{iv},$$

  where $Mov\_Token_A = \{ID_G \| ID_{A_i} \| ID_{A_v} \| ID_{M_i} \| A\_Key_v \| text\}$.

- To DKM, protected under the Domain-Area key DA_v_Key:

  $$AKM_v \rightarrow DKM : ID_{A_v} \| \{Mov\_Token_A \| text\} DA_v\_Key.$$

(d) If M_i is already on the MobList$_v$, AKM$_v$ will need to check whether there has been any re-keying of its area key since M_i's last visit to the area.

If there is none, AKM$_v$ sends a *move_welcome* message to M_i in the form of Mov_Token, along with its area key A_Key_v (as in *Step 4(c)*).

Otherwise, AKM$_v$ sends a *move_welcome* message to M_i in the form of Mov_Token, along with its current area key A_Key_{vnew}, where $Mov\_Token_A = \{ID_G \| ID_{A_i} \| ID_{A_v} \| ID_{M_i} \| A\_Key_{vnew} \| text\}$.

Step 5: Upon receiving the *move_welcome* message from AKM$_v$, DKM informs AKM$_i$ of the successful move of member M_i via a *move_welcome* message, protected under a *Domain-Area* key.

Note that other AKMs do not need to be notified of the member moving as it is not necessary for them to keep track of host mobility which occurs outside their areas.

Having the new information concerning member M_i, DKM and AKMs (AKM$_i$ and AKM$_v$) will need to update their *MobList* in order to keep track of member M_i's mobility, along with the number of area keys that may have been kept by M_i.

In summary, using this protocol, M_i moves from an area managed by an AKM$_i$ (where it is currently residing), to another visited area in a domain managed by AKM$_v$ while still remaining in the group session. The *move* is managed by the DKM via AKM$_i$. Assuming the establishment of the *session mobility key* between M_i and AKM$_v$ is successful, M_i is given the current area key of the visited area by AKM$_v$. All affected key managers during host mobility (such as DKM, AKM$_i$ and AKM$_v$) need to update their *MobList*, and new information regarding the moving member is logged into the lists.

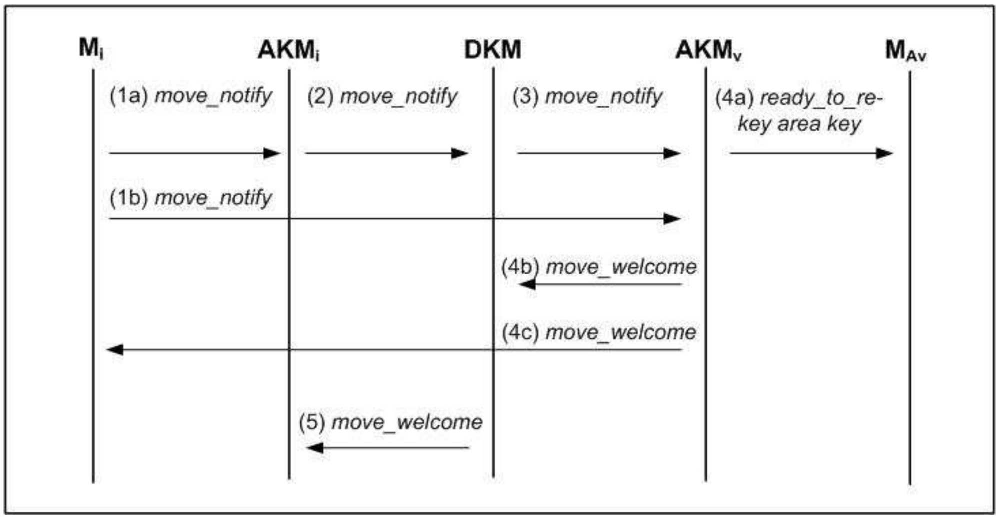


Figure 9.7: Member Moving with Backward Secrecy protocol message flow.

9.8 Protocol IV(b): Member Moving with Backward Secrecy

This protocol describes transfer of a group member from one area to another with consideration to secure access to the previous keys, as well as to past group data traffic (in other words, provision of backward secrecy). The protocol also includes the delivery of a new area key for the visited area A_Key_{vnew} to the moving member, as well as to the group members (if any) residing in that area.

Throughout this protocol we will make the same assumptions as in *Protocol IV(a)* (see *Section 9.7*). The message flow of this protocol is depicted in *Figure 9.7*, and to avoid being repetitive in demonstrating the steps involved in this protocol, only the steps that differ from *Protocol IV(a)* are described, as follows:

(a) In addition to *Step 4* in *Protocol IV(a)* (see *Section 9.7*), upon receiving the $move\_notify$ message from DKM and M_i, and assuming that both messages are valid, and that it is M_i's first time to enter the area (determined by AKM$_v$ checking that M_i is not in MobList$_v$), AKM$_v$ must re-key its area key A_Key_v. To do so, AKM$_v$ initiates *Protocol VI: Re-keying the area key* (see *Section 9.11*). This results in all group members M_{Av} in that particular area obtaining the new area key A_Key_{vnew}.

(b) AKM$_v$ then delivers the new area key to M_i, protected under a session mobility key $S_m\_Key_{iv}$ in the form of Mov_Token (as in *Step 4, Protocol IV(a)*) where $Mov\_Token_A = \{ID_G \| ID_{A_i} \| ID_{A_v} \| ID_{M_i} \| A\_Key_{vnew} \| text\}$:

$$AKM_v \rightarrow M_i : ID_{A_v} \| \{Mov\_Token_A \| text\} S_m\_Key_{iv}.$$

(c) Similarly to *Step 4(d)* in *Protocol IV(a)*, if M_i is already on the MobList$_v$, AKM$_v$ will need to check whether there has been any re-keying of its area key since M_i's last visit to the area. If none, AKM$_v$ sends a $move\_welcome$ message to M_i along with its current area key A_Key_v. Otherwise, AKM$_v$ sends an updated area key A_Key_{vnew} to M_i.

Note that, as for members moving without backward secrecy, other AKMs do not need to be notified of the member moving as it is not necessary for them to keep track of host mobility which occurs outside their areas. Also, re-keying of the area key due to host mobility only needs to occur within the visited area (which the member is moving to) and does not affect other areas.

In summary, using this protocol, M_i moves from an area managed by an AKM$_i$, to another visited area managed by AKM$_v$ while still remaining in the group session. For provision of backward secrecy, when a member moves to a visited area, the area key of visited area needs to be re-keyed. This results in group members (including the moving member) residing in that particular area obtaining a new area key. Similarly to *Protocol IV(a)*, all affected key managers during host mobility need to update their *MobList* with new information regarding the moving member.

We have provided two protocols for facilitating member moves, dependent on whether backward secrecy is required. These protocols feature a mechanism (MobList) that allows for efficient processing of members who are returning to recently visited areas.

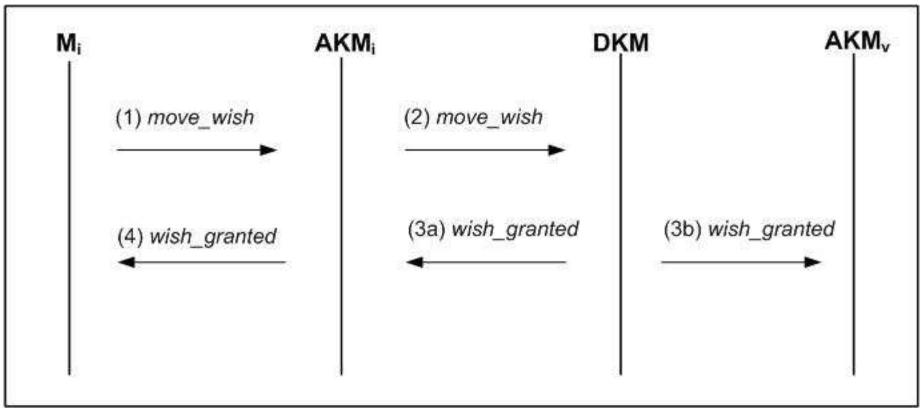


Figure 9.8: Establishment of a Session Mobility Key protocol message flow.

9.9 Protocol IV(c): Establishment of Session Mobility Key

This protocol describes establishment of a session mobility key to be used for host mobility between a moving member M_i and an area key manager of a visited area AKM$_v$. The protocol also includes the delivery of the session mobility key to the moving member M_i, as well as to the area key manager of a visited area AKM$_v$.

Throughout this protocol we will apply the same assumptions as those for *Protocol IV(a)* (see *Section 9.7*), which include the generic notation for describing the entities involved during host mobility.

The message flow of this protocol is depicted in *Figure 9.8*, and the steps involved are described as follows:

Step 1: A group member M_i who wishes to establish a session mobility key with an area key manager of a visited area AKM$_v$ sends a *move_wish* message to its area key manager AKM$_i$ (where M_i is currently residing), along with the ID of AKM$_v$ protected under $A_iM_i\_Key$ (see *Section 8.3.6.1* for initial key establishment):

$$M_i \rightarrow AKM_i : ID_{M_i} \| \{ID_G \| ID_{A_i} \| ID_{A_v} \| ID_{M_i} \| text\} A_iM_i\_Key.$$

Step 2: Upon receipt, AKM$_i$ checks the message by decrypting it with $A_iM_i\_Key$, which it shares with M_i, and passes the *move_wish* message to the DKM protected under DA_i_Key:

$$AKM_i \rightarrow DKM : ID_{A_i} \| \{ID_G \| ID_{A_i} \| ID_{A_v} \| ID_{M_i} \| text\} DA_i\_Key.$$

Step 3: Upon receipt, DKM does the following:

(a) It checks the message by decrypting it with DA_i_Key.

(b) Assuming that the checking is valid, DKM generates a session mobility key $S_m\_Key_{iv}$ (to be used between M_i and AKM$_v$).

(c) DKM then delivers a *wish_granted* message to AKM$_i$, along with the session mobility key in the form of $SKey\_Token_D$, where $SKey\_Token_D = \{ID_G \| ID_{A_i} \| ID_{A_v} \| ID_{M_i} \| S_m\_Key_{iv} \| text\}$, protected under DA_i_Key:

$$DKM \rightarrow AKM_i : ID_D \| \{SKey\_Token_D \| ID_D \| text\} DA_i\_Key.$$

(d) DKM also delivers a *wish_granted* message along with the session mobility key (in the form of $SKey\_Token_D$) to AKM$_v$, protected under DA_v_Key:

$$DKM \rightarrow AKM_v : ID_D \| \{SKey\_Token_D \| ID_D \| text\} DA_v\_Key.$$

Note that *Step 3c* and *3d* can take place independently and not necessarily in the order specified.

Step 4: Upon receiving the *wish_granted* message from the DKM, AKM$_i$ does the following:

(a) It checks the message by decrypting it with the DA_i_Key.

(b) AKM$_i$ then delivers the session mobility key to M_i in the form of $SKey\_Token_A$, protected under $A_iM_i\_Key$:

$$AKM_i \rightarrow M_i : ID_{A_i} \| \{SKey\_Token_A \| text\} A_iM_i\_Key,$$

where, $SKey\_Token_A = \{ID_G \| ID_{A_i} \| ID_{A_v} \| ID_{M_i} \| S_m\_Key_{iv} \| text\}$.

Step 5: Upon receiving the message from AKM$_i$, M_i checks the message by decrypting it with $A_iM_{i-}Key$ to obtain the session mobility key $S_{m-}Key_{iv}$.

Step 6: Upon receiving the *wish_granted* message from the DKM, AKM$_v$ checks the message by decrypting it with $DA_{v-}Key$ to obtain the session mobility key $S_{m-}Key_{iv}$, along with the ID of the moving member M_i with which it will share the session key.

Note that *Step 5* and *Step 6* can take place independently and not necessarily in the order specified.

We have provided a protocol for establishing a short-term session key for host mobility (in other words, to assist a member who wishes to move to another area). Like long-term keys (see *Section 8.3.6.1*), this session key is primarily used for establishing secure channels (see *Section 8.3.6.4*) between a moving member and an area key manager of a visited area. This key is valid throughout the member's residing period in the visited area, or until the member ceases to become a member of a multicast group.

9.10 Protocol V: Re-keying the Traffic Key

This protocol describes the re-keying of a traffic key that needs to occur during *Protocol II(b): New member joining with backward secrecy* (see *Section 9.4*), and *Protocol III(b): Existing member leaving with forward secrecy* (see *Section 9.6*). This protocol also includes the delivery of a new traffic key to group members, including the newly joined member (during *Protocol II(b)*), and to remaining members excluding the member who is leaving (during *Protocol III(b)*).

Throughout this protocol we make the same assumptions as in *Protocol II(b)* (see *Section 9.4*) and *Protocol III(b)* (see *Section 9.6*). The message flow of this protocol is depicted in *Figure 9.9*, and the steps involved are as follows:

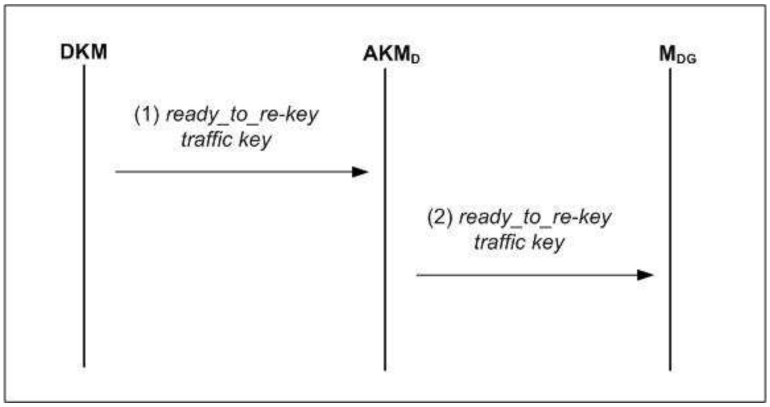

DKM
AKM$_D$
M$_{DG}$
(1) ready_to_re-key traffic key
(2) ready_to_re-key traffic key

Figure 9.9: Re-keying of a Traffic Key protocol message flow.

Step 1: In order to re-key the traffic key T_Key, DKM does the following:

(a) DKM generates a new traffic key T_Key_{new}.

(b) DKM then sends a $ready\_to\_re-key$ message, along with the new traffic key and the ID of the multicast group, to all area key managers AKM$_D$ in the form of $Rkey\_Token_D$, where $Rkey\_Token_D = \{ID_G \| T\_Key_{new} \| text\}$. DKM can send this message either by:

- *unicast*, where each message is protected under a *Domain-Area* key.
- *multicast*, where the message is protected under the domain key D_Key (see earlier protocols, such as those in *Section 9.2* and *Section 9.4*).

Step 2: Upon receiving the message, each AKM does the following:

(a) Checks the message by decrypting it with the key it shares with the DKM to obtain the new traffic key T_Key_{new}.

(b) Sends a $ready\_to\_re-key$ message to all group members in the area, along with the new traffic key in the form of $Rkey\_Token_A$, where $Rkey\_Token_A = \{ID_G \| ID_{A_i} \| T\_Key_{new} \| text\}$. AKM can send this message either by:

- *unicast*, where AKM sends the message to each member separately, protected under either an *Area-Member* key or a *session mobility* key for mobile members.

- *multicast*, where AKM sends a single message to all group members protected under the area key of an area.

Note that in *Protocol III(b)* for members leaving with forward secrecy, the delivery of a new traffic key to the remaining group members may need to be done via *unicast* to exclude the leaving member.

Step 3: Upon receiving the message from an AKM, a group member checks the message by decrypting it with the key it shares with the AKM to obtain the new traffic key T_Key_{new}.

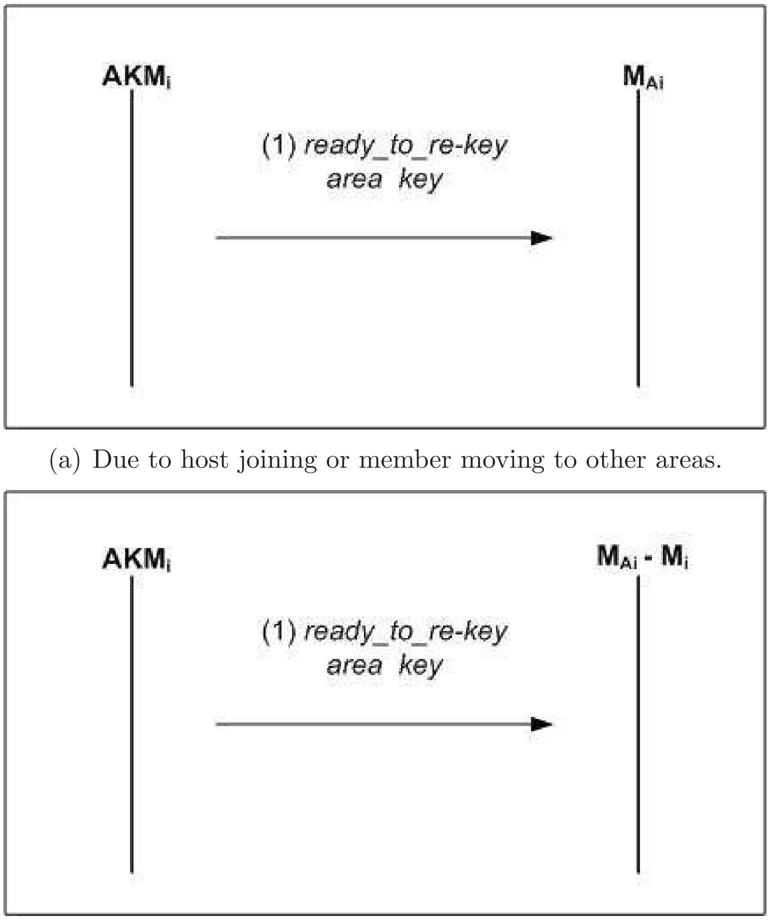
Note that in *Protocol II(b)* for new host joining, the delivery of a new traffic key T_Key_{new} can be delayed until after the re-keying of the area key has taken place, so that the AKM can use the area key to support secure distribution of the new traffic key to the group members in the area (in other words, via *multicast*).

We have provided the re-keying of the traffic key protocol that may need to run whenever there is a change in group membership.

9.11 Protocol VI: Re-keying the Area Key

This protocol describes the re-keying of an area key that needs to occur during *Protocol II(b): New member joining with backward secrecy* (see *Section 9.4*), *Protocol III(b): Existing member leaving with forward secrecy* (see *Section 9.6*), and *Protocol IV(b): Member moving with backward secrecy* (see *Section 9.8*). This protocol also includes the delivery of a new area key to group members, including the newly joined member (during *Protocol II(b)*) and the moving member (during *Protocol IV(b)*), as well as to remaining members excluding the member who is leaving (during *Protocol III(b)*).

Like *Protocol V* (see *Section 9.10*), we will make the same assumptions as in *Protocol II(b)* and *Protocol III(b)*, as well as making the assumptions of



(a) Due to host joining or member moving to other areas.

(b) Due to existing member leaving.

Figure 9.10: Re-keying of an Area Key protocol message flow.

Protocol IV(b) (see *Section 9.8*). The message flow of this protocol is depicted in *Figure 9.10*, and the steps involved are as follows:

Step 1: In order to re-key the area key A_Key, an AKM does the following:

(a) AKM generates a new area key A_Key_{new}.

(b) AKM then sends a $ready\_to\_re-key$ message along with the new area key A_Key_{new} to all group members in that area in the form of $Rkey\_Token_A$, where $Rkey\_Token_A = \{ID_{A_i} \| A\_Key_{new} \| text\}$. As in *Step 2(b) Protocol V* (see *Section 9.10*), AKM can send this message either via *unicast* or *multicast*.

Note that for *Protocol II(b)* and *Protocol IV(b)* (which is associated with new host joining and member moving to other areas), an AKM can use the old area key A_Key to secure the distribution of the new area key A_Key_{new} to group members in that area (see *Figure 9.10(a)*).

On the other hand, this may no longer apply for *Protocol III(b)* (existing member leaving protocol). In this case an AKM will need to send the new area key A_Key_{new} to group members separately, each of which is protected under an *Area-Member* key, in order to exclude the leaving member (see *Figure 9.10(b)*).

Step 2: Upon receiving the message from the AKM, each member M checks the message by decrypting it with the key it shares with the AKM to obtain the new area key A_Key_{new}.

We have provided the re-keying of area key protocol that may need to run whenever there is a change in group membership, or when a member moves to another area in a domain.

9.12 Summary

In this chapter, we specified the main GKMF protocols for our proposed framework. We have also introduced the use of lists as part of our protocol designs.

In the next chapter we provide the analysis of our proposed GKMF.

Chapter 10

GKMF: Analysis of the Proposal

This chapter provides basic analysis of the proposed GKMF specified in the previous chapters. This assesses whether the proposed framework meets the necessary requirements for secure deployment of multicast group communications in wireless mobile environments (WMobEs) that we identified in Chapter 8.

In this chapter we conduct a preliminary analysis of the proposed GKMF. This is a paper analysis against the requirements identified in *Section 8.2*. A full analysis of performance and scalability of the framework can probably only be verified through practical implementation (see *Chapter 11*). We believe however that this preliminary assessment provides useful results on the framework design.

We divide the assessment of the framework into three layers of analysis:

- **General analysis** (*Section 10.1*). The first layer analyzes the proposed framework as a whole, including whether it satisfies the general requirements (see *Section 8.2*).
- **Security analysis** (*Section 10.2*). The second layer covers security analysis of the framework. We divide this further into two parts. The first part covers a general security assessment as to whether it satisfies the requirements specified (see *Section 8.2.2*). The second part covers analysis of each of the proposed protocols identified within the framework (see *Chapter 9*), including whether it satisfies the functionalities requirements (see *Section 8.4*).

- **Performance analysis** (*Section 10.3*). The third layer covers analysis on performance and scalability of the framework as a whole, pertaining to the performance requirements specified in *Section 8.2.3*. This paper analysis includes assessment of costs in terms of:

  (a) *Operational complexity.* Measured by the number of encryptions or decryptions necessary during a group operation.

  (b) *Re-keying complexity.* Measured by the number of re-keying necessary during group operations.

  (c) *Storage complexity.* Measured by the number of keys each entity needs to store.

  (d) *Communication complexity.* Measured by the number of messages sent by entities involved in a group operation.

10.1 General Analysis

In this section, we analyze the proposed GKMF as a whole to see whether the proposed framework (including its components) meets the requirements specified in *Section 8.2*.

10.1.1 General Assessment

This analysis concerns the general requirements specified in *Section 8.2.1*.

(a) *Options for group membership policy*
Our GKMF supports dynamic membership policies, which can be managed using *Protocol I: Creation of New Group and initial distribution of keys* (see *Section 8.4.1* and *Section 9.2*).

(b) *Provision of host mobility*
One important feature of our proposal (as well as one of the main objectives of this thesis) for group communication in WMobEs is to specify a protocol

for group member moves to other areas. We have provided this as part of the protocol designs (see *Section 8.4.4*, *Section 9.7* and *Section 9.8*), including the option for provision of backward secrecy.

(c) *Reliable and trustworthy key managers*
We have assumed that all key manager entities (at the domain and area levels) within the proposed framework are trusted (see *Section 8.3.5*). These provide a secure foundation for designing the secure group key management services for group communication.

(d) *Scalability*
Our framework has several features that are designed to support scalability:

- The general architecture adopts a *hybrid* design approach (see *Section 4.2*) consisting of two levels (domain and area) of key managers. Each level is independently governed by a key manager. As the number of multicast groups increases, as well as the number of group members, additional AKMs can be added to support larger group operations.
- Re-keying due to group membership change is contained. In general, scalability problems are reduced by designing the architecture in such a way that any changes in group membership in a particular area do not go beyond that area, and other areas are not affected by the change. For example, during the new member joining protocol (with provision for backward secrecy) only the area key where the new join occurs needs to be re-keyed.

10.2 Security Analysis

In this section we analyze security of the proposed framework. We divide the assessment into two parts. *Section 10.2.1* provides a general security analysis of the whole framework. In the remaining sections (from *Section 10.2.2* onwards), we provide analysis each of the proposed protocols identified within the framework (see *Chapter 9*).

Recall that throughout the protocol designs, we made the following assumptions:

(a) Availability of secure encryption algorithms.

(b) Use of secure key establishment techniques to establish long-term keys.

(c) Use of secure entity and data origin authentication mechanisms to extend simplified protocols, discussed in *Chapter 9*.

(d) Use of some form of *time variant parameter* such as a time stamp in the *text* field within protocol messages for checking that a message received is not a replay of previous ones, as discussed in *Section 9.1.5*.

(e) The key managers (DKM and AKMs) in a domain are fixed and have been securely established prior to commencement of any multicast group communication, and every AKM has established a long-term *Domain-Area* key and a common domain key D_Key with the DKM (see *Section 8.3.6.1*).

(f) All keys managers (DKM and AKMs in a domain) as specified in *Section 8.3.2* are trusted entities which all group members trust.

(g) Availability of secure storage of cryptographic keys for all group communication entities (see *Section 8.3.6.3*).

(h) Availability of secure mechanisms for managing the *lists*: *HisList*, and *MobList*, as specified in *Section 9.1.4*.

As mentioned in *Section 9.1.3*, we do not discuss details of these here because their provision relies on generic techniques that are not specific to multicast group communications.

Throughout analysis each of the proposed protocol (from *Section 10.2.2* onwards), we have assumed that *freshness* of messages received is provided using some forms of *time variant parameter* such as a time stamp. Thus, if an adversary intercepts and later re-sends the message with an old time stamp, the intended recipient of the data would know that the time stamp received is out-of-bounds and not valid at that particular moment in time. If that is the case, the recipient of the data can conclude that the message received is no longer guaranteed as fresh, and the message can be discarded.

10.2.1 General Security Assessment

In this section we analyze the security of the proposed GKMF in general to see that the proposed framework meets the requirements specified in *Section 8.2.2*.

(a) *Provision of entity authentication*
We have assumed the use of secure entity authentication mechanisms in our proposed protocols (see *Section 9.1.3*). This provides a means for both group members and key manager entity(s) to authenticate and verify each other's identities.

(b) *Provision of backward and/or forward secrecy*
A particular security service that is specific to multicast group communication is the provision of confidentiality with respect to backward and forward secrecy. As mentioned in *Section 4.7.3*, a good design of GKMFs for secure multicast group communications should offer this as a default option. Our framework has provided these options for backward and/or forward secrecy, which can be managed using *Protocol I: Creation of new Group and initial distribution of keys* (see *Section 8.4.1*). Also, we have provided separate protocols for these options in new member joining, existing member leaving and member moving protocols (see *Chapter 9*).

(c) *Data (Message) integrity and authentication*
We have assumed the use of data origin authentication (hence data integrity) mechanisms in our proposed protocols (see *Section 9.1.3*). This provides a means for both group members and key manager(s) to verify the integrity of data received.

(d) *Secure data exchange*
Our framework supports secure data exchange, which can be achieved

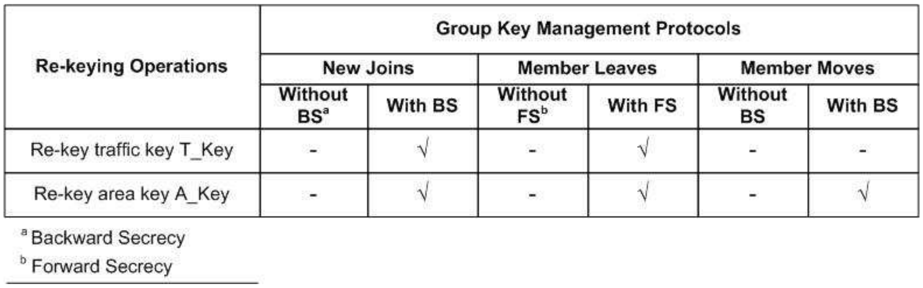
through careful application of security techniques and mechanisms, as discussed in *Section 9.1.2*. We have assumed secure use of these techniques and mechanisms, and the availability of secure encryption algorithms (as mentioned earlier, see *Section 10.2*). These provide a means for all group communicating entities to ensure that their communications remain confidential, and that access is only allowed to authorized group members.

(e) *Secure key distribution*
Our framework supports secure distribution of the keys needed in the framework. We have assumed that the distribution of long-term keys to key managers and group members (*Section 8.3.6.1*) is done in a secure manner, as discussed in *Section 8.3.6.3*. This provides a means for key managers to protect the distribution of short-term keys (*Section 8.3.6.2*) to all group members prior to group communication.

(f) *Secure key updates (re-keying)*
Our framework supports re-keying of short-term keys (in particular the traffic key of a multicast group and the area key of a particular area), which may need to occur whenever there is a change in group membership (either due to new joins, member leaves, or member moves). We have provided re-keying of these keys, which can be managed using *Protocol V: Re-keying the traffic key* (*Section 9.10*) and *Protocol VI: Re-keying the area key* (*Section 9.11*).

We have also provided options for re-keying to occur if provision for backward and/or forward secrecy is required. These can be managed using *Protocol II(b): New member joining with backward secrecy* (*Section 9.4*), *Protocol III(b): Existing member leaving with forward secrecy*(*Section 9.6*), and *Protocol IV(b): Member moving with backward secrecy* (*Section 9.8*).

We have assumed secure establishment of long-term keys between key managers and group members (*Section 8.3.6.1*), the use of secure entity and data origin authentication mechanisms in the protocols as discussed in *Chapter 9*, and the availability of secure encryption algorithms (as mentioned earlier, see *Section 10.2*). These provide a means for key managers to protect re-keying that needs to occur at the domain and area levels, and to securely inform group members when re-keying of a particular key is required.



Re-keying Operations	Group Key Management Protocols					
	New Joins		Member Leaves		Member Moves	
	Without BS[a]	With BS	Without FS[b]	With FS	Without BS	With BS
Re-key traffic key T_Key	-	√	-	√	-	-
Re-key area key A_Key	-	√	-	√	-	√

[a] Backward Secrecy

[b] Forward Secrecy

Table 10.1: Re-keying of traffic key and area key.

(g) *Additional key management during host mobility*

Our framework supports host mobility. We have provided the establishment of short-term *session mobility key* that needs to occur prior to host mobility using *Protocol IV(c): Establishment of session mobility key* (*Section 9.9*). Also, we have demonstrated that in order to preserve backward secrecy, a re-keying operation must occur whenever a member moves to another area, which can be managed using *Protocol IV(b): Member moving with backward secrecy* (*Section 9.8*).

Table 10.1 summarizes re-keying operations of both a traffic key T_Key and an area key A_Key, which occurred due to group membership change. We indicate re-keying of each key with a $\surd$ notation, otherwise they are indicated with a *dash*. From the table, with the exception of member moving, provision of backward and forward secrecy require re-keying of both T_Key and A_Key.

Re-keying of T_Key does not need to occur during host mobility because moving members are still in a same group session. When a member moves into an area, the A_key of the visited area (where the member is moving to) is re-keyed with a new area key.

(h) *Provision of trust model*

We have assumed the availability of secure environments for key management, as discussed in *Section 8.3.6.3*, and the existence of trustworthy and reliable key managers in a domain, as specified in *Section 8.3.2*. These provide a secure foundation for our framework to exercise the proposed group key management protocols for secure multicast group communication.

10.2.2 Protocol I: Creation of New Group and Initial Distribution of Keys

In this protocol, a new multicast group is created and initial distribution of a traffic key *T_Key* and an area key *A_Key* is conducted for all AKMs in a domain, and to the first member of a multicast group. The requirements for this protocol were identified in *Section 8.4.1*, and the protocol specified in *Section 9.2*.

We analyze the protocol as follows:

(a) Any host *M* who wishes to create a multicast group must first establish a secret *Area-Member* key with an AKM, and we have assumed that this was done securely (see *Section 8.3.6.1*).

(b) We have implicitly assumed that data origin authentication is provided by using a MAC. Thus, we can conclude that if an adversary wants to masquerade or initiate a bogus multicast group, the adversary will not be able to do so unless he has access to the MAC key. In the event that the MAC value received is not the same as the value a member computes, the message will be discarded. The same process applies to the key managers involved (DKM and AKM).

(c) The host *M* uses the *Area-Member* key for protecting the communications between itself and the AKM. If an adversary gets hold of the encrypted messages between *M* and AKM, the adversary has no way of decrypting the messages because he does not have access to the *Area-Member* key.

(d) After granting the permission to create a multicast group, DKM generates and distributes a traffic key *T_Key* to all AKMs. The distribution of this key to all AKMs is protected either by the *Domain-Area* key (if the key is to be sent separately via unicast to every AKM), or a common domain key *D_Key* (if the key is to be sent one time via multicast). If an adversary wants to get hold of the *T_Key*, the adversary has no access to either of these keys (*Domain-Area* or *D_Key*), so he cannot obtain the *T_Key*.

(e) Similarly, an AKM (when the host joins a multicast group) generates and distributes an area key A_Key to the host M along with the T_Key it receives from the DKM. The distribution of these keys is protected under the *Area-Member* key which is shared only between the AKM and host M. The adversary has no access to the *Area-Member* key, so he cannot obtain the T_Key or the A_Key.

(f) Other information distributed during this protocol is also protected under secret keys known only to key managers (DKM and AKMs) and host M. Thus, a passive observer knows nothing about the properties of the new multicast group.

10.2.3 Protocol II(a): New Member Joining without Backward Secrecy

In this protocol, a new join of a host to become a member of a multicast group is conducted with no provision of backward secrecy. This means that when a new member joins a multicast group, the same keys (T_Key and A_Key) that are currently in use are given to the newly joined member. We have assumed that *Protocol I* (*Section 9.2*) was successfully conducted. The requirements for this protocol were identified in *Section 8.4.2* and the protocol specified in *Section 9.3*.

We analyze the protocol as follows:

(a) Any host M who wishes to join a multicast group must first establish a secret *Area-Member* key with an AKM, and we have assumed that this was done securely (see *Section 8.3.6.1*).

(b) As for (b) in *Section 10.2.2*, we have implicitly assumed that data origin authentication is provided by using a MAC. Thus, we can conclude that if an adversary wants to masquerade as someone else to join a multicast group, the adversary will not succeed as he has no access to the MAC key and cannot produce the same MAC value. In the event that the MAC value received is not the same as the value a member computes, the message will be discarded. Other entities (DKM and AKM) can check the integrity of the message received via the same process.

(c) As for (c) in *Section 10.2.2*, the host M uses the *Area-Member* key for protecting the communications between itself and the AKM. If an adversary gets hold of the encrypted messages between M and AKM, the adversary has no way of decrypting the messages because he does not have access to the *Area-Member* key.

(d) After receiving the $join\_request$ from M, AKM relays the request to the DKM protected under the *Domain-Area* key. If an adversary gets hold of the encrypted messages between AKM and DKM, the adversary has no way of decrypting the messages because he does not have access to the *Domain-Area* key shared only between AKM and DKM.

(e) After receiving the $join\_granted$ message from DKM, the AKM sends the current keys to M in the form of $Join\_Token$. This message is protected under an *Area-Member* key shared only between AKM and M. If an adversary wants to get hold of the token, the adversary has no access to the *Area-Member* key, so he cannot obtain T_Key, A_Key or other group-related information. If an adversary intercepts or modifies the message content, this can easily be detected by DKM, AKM or M when the implicit MAC value is checked against the value received.

10.2.4 Protocol II(b): New Member Joining with Backward Secrecy

In this protocol, a new join of a host to become a member of a multicast group is conducted with provision of backward secrecy. When a new member joins a multicast group, re-keying of cryptographic keys occurs. This results in the new member and other members in the area (where the new join occurs) obtaining new keys T_Key_{new} and A_Key_{new}. This also results in other group members across the domain obtaining a new T_Key_{new}. The requirements for this protocol were identified in *Section 8.4.2* and the protocol specified in

Section 9.4.

As for *Protocol II(a) Section 10.2.3*, we have assumed that *Protocol I* (*Section 9.2*) was successfully conducted. Due to similarity with *Protocol II(a)*, we only analyze the differences, as follows:

(a) After receiving the $join\_granted$ message from DKM, AKM initiates the re-keying of its area key A_Key, which results in all members in the area (including the newly joined member) obtaining a new area key A_Key_{new}. AKM can send this new area key (in the form of $Join\_Token$, as in *Section 10.2.3*) to the existing members in the area under protection of the old area key A_Key (via *multicast*), or under protection of *Area-Member* keys (via *unicast*). AKM sends this key to the new member protected under an *Area-Member* key, and we have assumed that this was done securely (see *Section 8.3.6.1*). If an adversary wants to get hold of A_Key_{new}, then as the adversary has no access to either of these keys (*Area-Member* key, or A_Key), he cannot obtain A_Key_{new}.

(b) After granting the host permission to join the multicast group, DKM initiates the re-keying of the group's traffic key T_Key, which results in all AKMs and all group members (via AKM) in the domain obtaining a new traffic key T_Key_{new}. DKM can send this new traffic key to all AKMs under protection of the domain key D_Key (via *multicast*), or under protection of the *Domain-Area* keys (via *unicast*). If an adversary wants to get hold of T_Key_{new}, then as the adversary has no access to either of these keys (*Domain-Area* or D_Key), he cannot obtain T_Key_{new}.

(c) The delivery of T_Key_{new} to group members across the domain is done by each AKM governing an area. Similarly to (a), AKM can *multicast* this T_Key_{new} to members in the area protected under A_Key, or *unicast* under the *Area-Member* keys. If an adversary gets hold of the encrypted messages between AKM and the group members, the adversary has no way of decrypting the messages because he does not have access to the secret shared only between AKM and M, or the area key A_Key.

(d) AKM can delay the delivery of A_Key_{new} to the newly joined member until it receives T_Key_{new} from the DKM. AKM can send both keys (in the form of $Join\_Token$) under the protection of the *Area-Member* key, and we have assumed that this was done securely.

10.2.5 Protocol III(a): Existing Member Leaving without Forward Secrecy

In this protocol, an existing member leaving a multicast group is conducted with no provision of forward secrecy. This means that when a member leaves a multicast group it requires no further processing. A member is excluded from a group and the reason of leaving is logged in *HisList*. We have assumed that there is an established multicast group (*Section 9.5*). The requirements for this protocol were identified in *Section 8.4.3* and the protocol specified in *Section 9.5*.

We analyze the protocol as follows:

(a) Any member M wishing to leave a multicast group sends a $leave\_notify$ message to AKM protected under an *Area-Member* key, who then passes the message to the DKM protected under a *Domain-Area* key, and we have assumed that these keys were established securely between the member M and AKM, and between AKM and DKM (see *Section 8.3.6.1*).

(b) As in (b) *Section 10.2.2*, we have implicitly assumed the provision of data origin authentication using MACs. Thus, we can conclude that if an adversary wants to masquerade as someone else in order to leave a multicast group (or to send an $eject\_notify$ message), the adversary will not able to do so as he has no access to the MAC key. Even if the adversary could forge a key and produce a MAC, this can be easily detected by the managing entities (such as DKM and AKM) when the MAC value they produce (using the correct key) is not the same as the value obtained from the message (using the forged key). Through this process, a DKM or an AKM governing the *leave* operations can conclude that integrity of the $leave\_notify$ message received is no longer guaranteed, and the message can be discarded.

(c) As in (c) *Section 10.2.2*, the member M uses the *Area-Member* key for protecting the communications between itself and the AKM. If an adversary gets hold of the encrypted messages between the host M and AKM, the adversary has no way of decrypting the messages because he does not have access to the *Area-Member* key. Likewise with the *unicast* communications between AKM and DKM, which are protected under the *Domain-Area* key. The adversary has no access to this key, hence cannot decrypt the encrypted messages between them.

(d) After receiving the $leave\_notify$ message from AKM, DKM updates its *HisList*, and the reason for leaving is logged. We have assumed that this list is maintained and kept securely by the DKM.

10.2.6 Protocol III(b): Existing Member Leaving with Forward Secrecy

In this protocol, an existing member leaving a multicast group is conducted with provision of forward secrecy. When a member leaves, the remaining members of the multicast group need to be re-keyed. This results in all remaining group members in an area where the leave occurs obtaining a new area key A_Key_{new}, and all AKMs and group members in the domain obtaining a new traffic key T_Key_{new}. The requirements for this protocol were identified in *Section 8.4.3* and the protocol specified in *Section 9.6*.

As in *Protocol III(a)* (*Section 10.2.5*), the information about the leaving member is logged in *HisList*. Due to the similarity with *Protocol III(a)*, we just analyze the differences, as follows:

(a) After receiving the $leave\_notify$ message from M (or an $eject\_notify$ message from DKM), AKM initiates the re-keying of its area key A_Key. This results in all remaining group members in the area (excluding the leaving member) obtaining a new area key A_Key_{new}. This new key is sent via *unicast*, protected under the *Area-Member* keys. If an adversary gets hold of the encrypted message, he will not be able to decrypt it as he has no access to the secret shared only between each member and AKM.

(b) After receiving the $leave\_notify$ message from AKM (or after sending an $eject\_notify$ to AKM), DKM initiates the re-keying of the group's traffic key T_Key. This results in all AKMs and group members (via AKM) in the domain obtaining a new traffic key T_Key_{new}. As in *Protocol II(b)* (see *Section 10.2.4*), DKM can send this new key to all AKMs either via *multicast* protected under D_Key, or via *unicast* protected under *Domain-Area* keys. If an adversary wants to get hold of this new key, he will not be able to do so because he has no access to either of the keys (D_Key, or *Domain-Area* keys) used to protect the new key.

(c) As in *Section 10.2.5*, we have implicitly assumed the provision of data origin authentication using MACs. We can conclude that if an adversary wants to masquerade as someone else in order to leave a multicast group (or to eject a member), this can be easily detected by the managing entity (DKM or AKM) when the MAC value computed differs from the value obtained from the received message. Likewise with the group member who might receive a false $eject\_notify$ from an alleged DKM or AKM. In either case, DKM, AKM or M can conclude that the integrity of the message received is no longer guaranteed, and the message can be discarded without further processing.

10.2.7 Protocol IV(a): Existing Member Moving without Backward Secrecy

In this protocol, the transfer of a group member from one area to another is conducted with no consideration for backward secrecy. This means that when a member moves from one area to another, the member is given the area key A_Key_v of the visited area. All key managers (DKM and all AKMs) in the domain need to update their *MobList* whenever a *move* occurs. We have assumed that these lists are maintained and kept securely by the key managers (see *Section 9.1.4*). We have also assumed that there is an established multicast group (see *Section 9.7*). The requirements for this protocol were identified in *Section 8.4.4* and the protocol specified in *Section 9.7*.

We analyze the protocol as follows:

(a) A member M_i who wishes to move into another area must first establish a short-term *session mobility* key with the AKM of the visited area, and we have assumed that this was done securely (see *Section 9.7*).

(b) A member M_i uses this short-term key to secure communications with the AKM of the visited area. If an adversary wants to masquerade as some moving member in order to get hold of the area key A_Key_v of the visited area, he will not be able to do so because he has no access to the *session mobility* key shared only between the moving member and the AKM of the visited area.

(c) We have implicitly assumed the provision of data origin authentication using MACs. Thus, we can conclude that if an adversary wants to masquerade as some moving member in order to get hold of A_Key_v, the adversary will not able to do so because he has no access to the MAC key. Other entities (DKM, AKM and M_i) can easily check the integrity of messages received via the same process.

(d) After obtaining the *session mobility* key, M_i initiates the *move* protocol by sending a $move\_notify$ message to its local area key manager AKM$_i$, protected under the *Area-Member* key, and to the visited area key manager AKM$_v$, protected under the *session mobility* key. If an adversary gets hold of the enciphered messages between the entities, he has no way of deciphering the message as he has no access to either of the keys (*Area-Member* key, or *session mobility* key).

(e) After receiving the $move\_notify$ message from M_i, AKM$_i$ passes the message to DKM, protected under the *Domain-Area* key. As in (d), if an adversary gets hold of the enciphered message between DKM and AKM$_i$, he has no way of deciphering the message as he has no access to the *Domain-Area* key.

(f) After receiving the $move\_notify$ message from AKM$_i$, DKM notifies AKM$_v$ of the move (in the form of a token, see *Section 9.7*), protected under the *Domain-Area* key. Similarly, if an adversary gets hold of the enciphered message between DKM and AKM$_v$, he has no way of deciphering the message as he has no access to the *Domain-Area* key.

(g) After receiving the $move\_notify$ message from DKM, AKM$_v$ acknowledges the move by M_i and sends its area key A_Key_v (A_Key_{vnew} is sent if there has been re-keying of its area key) to M_i protected under the *session mobility* key. If an adversary wants to get hold of A_Key_v or A_Key_{vnew}, he will not be able to do so because he has no access to the *session mobility* key.

(h) All affected key managers (DKM, AKM$_i$ and AKM$_v$) update their *MobList*, and area(s) visited are logged. We assume that these lists are maintained and kept securely by the key managers.

10.2.8 Protocol IV(b): Member Moving with Backward Secrecy

In this protocol, the transfer of a group member from one area to another is conducted with provision for backward secrecy. When a member moves from one area to another, the area where the member is moving to (visited area) needs to be re-keyed with a new area key. This results in all group members in the visited area, including the moving member, obtaining a new area key A_Key_{vnew}. The requirements for this protocol were identified in *Section 8.4.4* and the protocol specified in *Section 9.8*.

As in *Protocol IV(a) Section 10.2.7*, the information on the member moved is logged in each affected key manager's *MobList*. Due to similarity with *Protocol IV(a)*, we only analyze the differences, as follows:

(a) After receiving the $move\_notify$ message from DKM and M_i (and if M_i is not in the *MobList*), AKM$_v$ initiates the re-keying of its area key A_Key_v.

This results in all members residing in the visited area, including the moving member M_i, obtaining the new area key A_Key_{vnew}. AKM$_v$ can send this key to group members (excluding M_i) in the area either via *multicast*, protected under the old area key A_Key_v, or via *unicast*, protected under the *Area-Member* keys. AKM$_v$ sends A_Key_{vnew} to M_i (in the form of $Move\_Token_A$, see *Section 9.8*), protected under the *session mobility* key. If an adversary wants to get hold of A_Key_{vnew}, he will not be able to do so because he has no access to the keys (A_Key_v, *Area-Member* keys, or *session mobility* key).

(b) We have implicitly assumed the provision of data origin authentication, so we can conclude that if an adversary wants to masquerade as some moving member in order to get hold of A_Key_{vnew}, the adversary will not be able to do so as he has no access to the MAC keys.

(c) On a member's first move into an area, the area needs to be re-keyed with a new area key, and information about the moving member is logged in *MobList*. If it is necessary to control the number of area keys that are kept by a group member (which corresponds to the number of areas that he visited), *MobList* may need to be reset for that particular member after a period of time, for example when the number of area keys collected by a member (as he moves from one area to another) has reached a threshold limit. In this case, re-keying of the area key may need to occur when the member moves into an area. This will be determined by the group security policy at the creation of a multicast group, prior to the commencement of group communication. This is useful to avoid a group member moving from one area to another with intent to collect all the area keys. If colluding members want to exchange security information, such as area keys, to gain unauthorized access to different areas, this could also be prevented (periodically).

10.2.9 Protocol IV(c): Establishment of Session Mobility Key

In this protocol, a *session mobility* key for *host mobility* is established between the moving member M_i and the AKM of the visited area AKM$_v$. This results in M_i and AKM$_v$ obtaining the session mobility key $S_{m-}Key_{iv}$. The requirements for this protocol were identified in *Section 8.4.4* and the protocol specified in *Section 9.9*.

As part of *Protocol IV(a)* and *Protocol IV(b)* (see *Section 10.2.7* and *Section 10.2.8*), we have assumed that there is an established multicast group.

We analyze the protocol as follows:

(a) A member M_i who wishes to establish a *session mobility* key with AKM$_v$, must first send a $move\_wish$ message to its local area key manager AKM$_i$. This message is protected under the *Area-Member* key shared only between M_i and AKM$_i$, and we have assumed that this was done securely (see *Section 8.3.6.1*).

(b) After receiving the $move\_wish$ message from AKM$_i$, DKM generates a *session mobility* key, and we have assumed that this was done securely (see *Section 8.3.6.3*).

(c) DKM sends this key to AKM$_i$ (in the form of $SKey\_Token_D$, see *Section 9.9*), and to AKM$_v$, each protected under a *Domain-Area* key. AKM$_i$ then sends the key to M_i (in the form of $SKey\_Token_A$) protected under an *Area-Member* key. If an adversary wants to get hold of the *session mobility* key, he will not be able to do so because he has no access to the keys (*Domain-Area* key, or *Area-Member* key).

(d) We have implicitly assumed the provision of data origin authentication, so we can conclude that if an adversary tampers with any part of the message contents, the group entities (such as AKM$_i$ and M) can easily check the integrity of the received message using a MAC value. The same process applies to other entities such as DKM and AKM$_v$.

10.2.10 Protocol V: Re-keying the Traffic Key

In this protocol, the re-keying of a traffic key T_Key is conducted. This protocol occurs during *Protocol II(b): Member joining with backward secrecy* (see *Section 9.4*) and *Protocol III(b): Member leaving with forward secrecy* (see *Section 9.6*). This results in all group members, including the newly joined member (during *Protocol II(b)*), and remaining members, excluding the member who is leaving (during *Protocol III(b)*), obtaining a new traffic key. The requirements for this protocol were identified in *Section 8.4.5* and the protocol specified in *Section 9.10*.

As we have analyzed these protocols, which include the re-keying of traffic key (see *Section 10.2.4* and *Section 10.2.6*), we just analyze the differences:

(a) To re-key a traffic key T_Key, DKM generates a new traffic key T_Key_{new}, and we have assumed that this was done securely (see *Section 8.3.6.3*).

(b) DKM sends the T_Key_{new} (in the form of $RKey\_Token_D$, see *Section 9.10*) to all area key managers AKM$_D$ protected either under D_Key (via *multicast*), or *Domain-Area* keys (via *unicast*).

(c) After receipt, each AKM sends the T_Key_{new} (in the form of $RKey\_Token_A$, see *Section 9.10*) protected either under A_Key (via *multicast*), or *Area-Member* keys and *session mobility* keys (for mobile members in the area) (via *unicast*). If an adversary wants to get hold of T_Key_{new}, he will not be able to do so as he has no access to the keys (D_Key, A_Key, *Domain-Area* keys, *Area-Member* keys, or *session mobility* keys). If an adversary manages to tamper with all or some parts of the messages, this can be easily detected by the group entities (DKM, AKM$_D$ and M) through application of the implicit MACs.

10.2.11 Protocol VI: Re-keying the Area Key

In this protocol, the re-keying of an area key A_Key is conducted. This protocol occurs during *Protocol II(b): Member joining with backward secrecy* (see *Section 9.4*), *Protocol III(b): Member leaving with forward secrecy* (see *Section 9.6*), and during *Protocol IV(b): Member moving with backward secrecy* (see *Section 9.8*). This results in group members including the newly joined member (during *Protocol II(b)*), remaining members excluding the member who is leaving (during *Protocol III(b)*), and the moving member (during *Protocol IV(b)*), obtaining a new area key. The requirements for this protocol were identified in *Section 8.4.5* and the protocol specified in *Section 9.11*.

As we have analyzed these protocols, which include the re-keying of area key (see *Section 10.2.4*, *Section 10.2.6* and *Section 10.2.8*) we only analyze the differences:

(a) To re-key an area key A_Key, AKM generates a new area key A_Key_{new}, and we have assumed that this was done securely (see *Section 8.3.6.3*).

(b) AKM sends the A_Key_{new} (in the form of $RKey\_Token_A$, see *Section 9.11*) to all group members in the area, protected either under an old area key A_Key (via *multicast*), or *Area-Member* keys (via *unicast*).

(c) AKM can use the old area key to delivery the A_Key_{new} via *multicast* (as mentioned in (b)) for *Protocol II(b)* and *Protocol IV(b)* (member joining and member moving). However, for *Protocol III(b)* (member leaving) AKM needs to use *unicast* protected under *Area-Member* keys to exclude the leaving member. If a leaving member wants to access the group traffic after he leaves the group, he will not be able to do so because he has no access to the new key.

10.3 Performance Analysis

This analysis is concerned with the performance requirements specified in *Section 8.2.3*. As mentioned earlier, the analysis on performance and scalability of the proposed framework is presented in terms of *operational complexity*, *re-keying complexity*, *storage complexity*, and *communication complexity*.

We use the following notation to analyze the performance of the protocols:

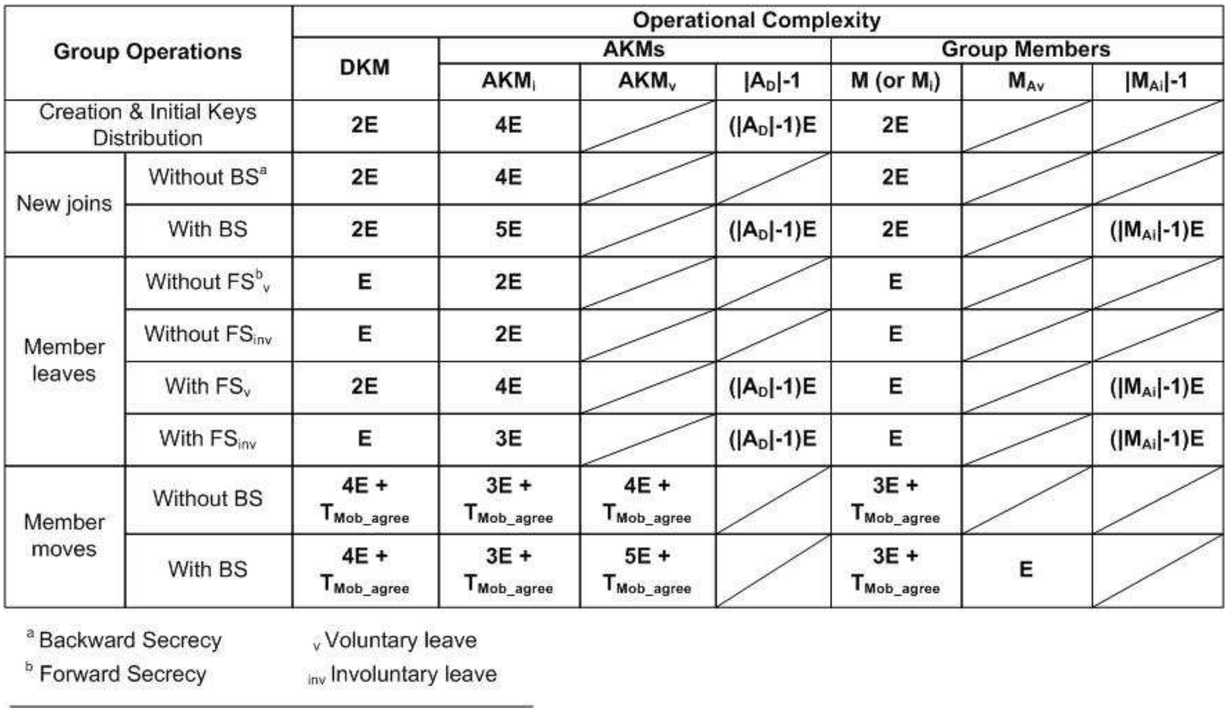
(a) Generic notation such as DKM, AKM and M (or M_i) to denote domain key managers, area key managers and group members of a multicast group as in earlier protocol designs are also used here (see *Section 9.1*). In addition:

- Let $|A_x|$ be the number of group members in an area x.
- Let $|A_D|$ be the number of areas in a domain D (and hence AKMs).
- Let $|TK_D|$ be the number of traffic keys in a domain.
- Let $|TK_A|$ be the number of traffic keys in an area.
- Let $|h_{Mob}|$ be the number of h_{Mob} in an area, where an h_{Mob} is a set of security parameters, consisting a *session mobility key* and an *area key* of a visited area, needed by a group member for *host mobility*.

The performance assessment of our basic protocol designs is categorized based on the costs incurred, as follows:

(a) *Operational complexity*
This assessment demonstrates the framework performance with respect to:

- the number of encryptions (or decryptions) that need to be performed during secure group operations. We note that *one encryption (or decryption) is equivalent to one cost*, and denote this by E.
- As host mobility may require additional key management, member moving protocols require additional operational costs with respect to establishing an h_{Mob}, prior to moving (between a member M_i and



Group Operations		Operational Complexity						
		DKM	AKMs			Group Members		
			AKM$_i$	AKM$_v$	$\lvert A_D \rvert$-1	M (or M$_i$)	M$_{Av}$	$\lvert M_{Ai} \rvert$-1
Creation & Initial Keys Distribution		2E	4E		($\lvert A_D \rvert$-1)E	2E		
New joins	Without BS[a]	2E	4E			2E		
	With BS	2E	5E		($\lvert A_D \rvert$-1)E	2E		($\lvert M_{Ai} \rvert$-1)E
Member leaves	Without FS[b]$_v$	E	2E			E		
	Without FS$_{inv}$	E	2E			E		
	With FS$_v$	2E	4E		($\lvert A_D \rvert$-1)E	E		($\lvert M_{Ai} \rvert$-1)E
	With FS$_{inv}$	E	3E		($\lvert A_D \rvert$-1)E	E		($\lvert M_{Ai} \rvert$-1)E
Member moves	Without BS	4E + T_{Mob_agree}	3E + T_{Mob_agree}	4E + T_{Mob_agree}		3E + T_{Mob_agree}		
	With BS	4E + T_{Mob_agree}	3E + T_{Mob_agree}	5E + T_{Mob_agree}		3E + T_{Mob_agree}	E	

[a] Backward Secrecy $\quad_v$ Voluntary leave

[b] Forward Secrecy $\quad_{inv}$ Involuntary leave

Table 10.2: Cost assessment based on operational complexity.

key managers (DKM, AKM$_i$ and AKM$_v$)). We note that *one* h_{Mob} *is equivalent to one cost*, and denote this by T_{Mob_agree}.

The assessment for all protocols is summarized in *Table 10.2*.

From *Table 10.2*, the main cost for each group operation can be seen as reasonably spread amongst the key managers (see *column* DKM and *column* AKM$_i$). As illustrated, the cost for the provision of backward and/or forward secrecy during new joins and leaves operations is slightly higher compared to when it is not required.

Importantly, group members are only required to perform minimal computations, which mostly involve one or two E during most of the group operations (see *column* M (or M_i)).

For example, member move protocols (for all entities including the group members) require additional costs, which is T_{Mob_agree} for establishing the h_{Mob}, prior to moving. The overall cost for member moves is considered reasonably minimal for a group member (see *column* M (or M_i)). This is suitable for members (i.e. mobile devices) in wireless mobile environments, which mostly have limited computation capabilities and limited battery power for more cryptographic computations. For example, a typical mobile device 206MHz processor with 64MB of RAM, powered by a 950 mAH rating Li-Polymer battery (Potlapally et al., 2006) can easily cope with the operational cost of $3E+T_{Mob-agree}$, incurred by a group member during member moves (see *column M* (or M_i)).

The use of symmetric encryption (as discussed in *Section 8.3.6*) is also an advantage as it is computationally faster, hence saves battery power.

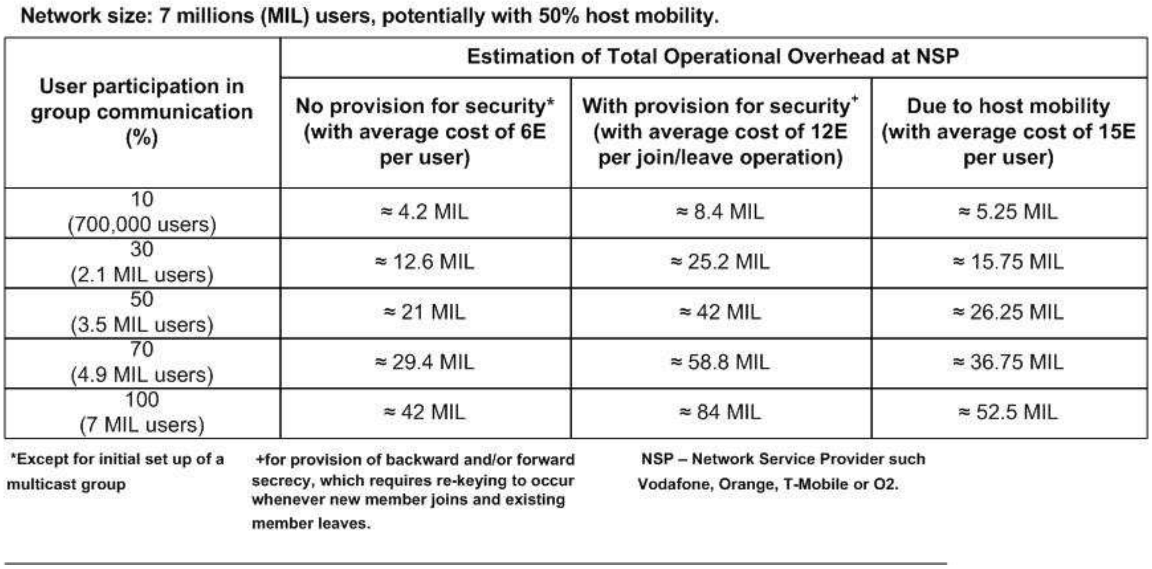
Table 10.3 illustrates an example of performance overhead that may incur in a larger scale of network size. (BBC, 2007) and (TechNews, 2007) report that at least 7 millions (MIL) people are anticipated to be using iPhone (the latest smart phone technology) (Apple, 2007) in UK by the end of 2008. Based on this information, we illustrate the example by using the same network size.

Thus, assuming that a network service provider (NSP) such as Vodafone, Orange, T-Mobile or O2 will have at least 7 millions iPhone users, each of which has the multicast capability to engage in group communication. With some degree of host mobility that may occur during the participation in a multicast group across multiple areas (which is assumed to be under the NSP's jurisdiction), *Table 10.3* provides some estimations in terms of performance overhead that an NSP will need to bear if secure group communication is to be deployed in WMobEs. The estimation cost provided is based on the E cost obtained from *Table 10.2*.

From *Table 10.3*, based on the network size of 7 MIL with 50% are potential for mobility, note that:

- The first column represents the number of user participation (%) in multicast group communication (in other words, number of people who use group-based services such as taking part in multimedia conferencing or getting special news updates).
- The other columns represent cost estimation in terms of performance overhead due to provision of security and host mobility.

  Note that the average cost listed in each performance overhead (see 2nd, 3rd and 4th columns) is obtained from *Table 10.2*. For example,



Network size: 7 millions (MIL) users, potentially with 50% host mobility.

User participation in group communication (%)	Estimation of Total Operational Overhead at NSP		
	No provision for security* (with average cost of 6E per user)	With provision for security+ (with average cost of 12E per join/leave operation)	Due to host mobility (with average cost of 15E per user)
10 (700,000 users)	≈ 4.2 MIL	≈ 8.4 MIL	≈ 5.25 MIL
30 (2.1 MIL users)	≈ 12.6 MIL	≈ 25.2 MIL	≈ 15.75 MIL
50 (3.5 MIL users)	≈ 21 MIL	≈ 42 MIL	≈ 26.25 MIL
70 (4.9 MIL users)	≈ 29.4 MIL	≈ 58.8 MIL	≈ 36.75 MIL
100 (7 MIL users)	≈ 42 MIL	≈ 84 MIL	≈ 52.5 MIL

*Except for initial set up of a multicast group

+for provision of backward and/or forward secrecy, which requires re-keying to occur whenever new member joins and existing member leaves.

NSP – Network Service Provider such Vodafone, Orange, T-Mobile or O2.

Table 10.3: An example of cost estimation on performance overhead.

> the average cost of 6E per user (in 2nd column) is the average of operational cost incurred by DKM and AKM$_i$ (see DKM and AKM$_i$ columns in the 1st row *Table 10.2*). Similarly, the average cost of 15E for host mobility (last column) is the average of operational cost incurred during member moves protocol in *Table 10.2* (see last row in DKM, AKM$_i$ and AKM$_v$ columns).

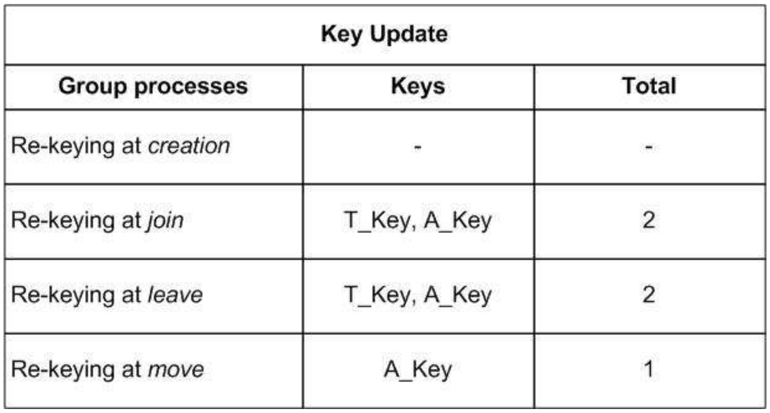
It shows that as the network size increases along with host mobility, as well as group membership, the amount of performance overhead (that the network has to manage) also increases. For example, with no provision of security, 10% user participation (700,000 out of 7 MIL users) requires overhead cost of $\approx$ 4.2 MIL, while $\approx$ 5.25 MIL may be needed for host mobility. Evidently, host mobility requires additional $\approx$ 1 MIL overhead cost to operate.

Although it seems that the amount of overhead for host mobility is quite high (even for 10% user participation), it is still reasonably small considering the network size of 7 MIL users.

With the advance of mobile technology and economic growth, wide spread deployment of secure group communication in WMobEs may be possible. For example, Vodafone UK (Vodafone, 2007) claims to have at least 20 MIL active users, and that it has established around 80% out of coverage areas in UK with at least 20,000 base stations. With such established infrastructures, we believe that it can comfortably cope with the amount of overhead that may be incurred by the network size of 7 MIL users as discussed.

(b) *Re-keying Complexity*

This assessment demonstrates the cost in terms of the number of key updates (or re-keying) that has to occur. Assessment is based on *one re-keying is equivalent to one cost*, and we summarize this in *Table 10.4*.



Key Update		
Group processes	**Keys**	**Total**
Re-keying at *creation*	-	-
Re-keying at *join*	T_Key, A_Key	2
Re-keying at *leave*	T_Key, A_Key	2
Re-keying at *move*	A_Key	1

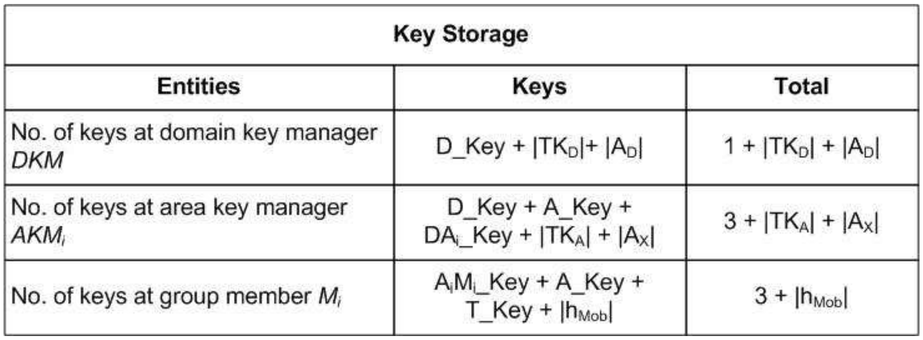
Table 10.4: Cost assessment based on re-keying complexity.

Table 10.4 shows that while there is no re-keying cost at creation of a multicast group, two key updates (re-keying of a traffic key T_Key and re-keying of an area key A_Key) are required every time a new join, or a leave occurs. There is no need for key update if provision of backward and forward secrecy is not required, thus no cost in terms of re-keying incurs.

On the other hand, a *move* requires only one re-keying cost, and that is for re-keying the area key of the visited area (where a member is moving to). Depending on requirements of multicast applications, the cost of re-keying can be reduced to half of the key updates normally required (as shown in *Table 10.4*), if only one provision for either backward or forward secrecy is required. For example, if provision of security is not required during host mobility (in other words, no backward secrecy) and group members are free to move between areas, no re-keying needs to occur, thus no cost incurs.

(c) *Storage complexity*

This assessment demonstrates the cost in terms of the amount of key storage required by communicating entities. Assessment is based on *one key stored is equivalent to one cost.* We summarize the cost of key storage at each entity in *Table 10.5*.



Key Storage		
Entities	**Keys**	**Total**
No. of keys at domain key manager *DKM*	D_Key + $|TK_D|$+ $|A_D|$	1 + $|TK_D|$ + $|A_D|$
No. of keys at area key manager *AKM$_i$*	D_Key + A_Key + DA$_i$_Key + $|TK_A|$ + $|A_X|$	3 + $|TK_A|$ + $|A_X|$
No. of keys at group member *M$_i$*	A$_i$M$_i$_Key + A_Key + T_Key + $|h_{Mob}|$	3 + $|h_{Mob}|$

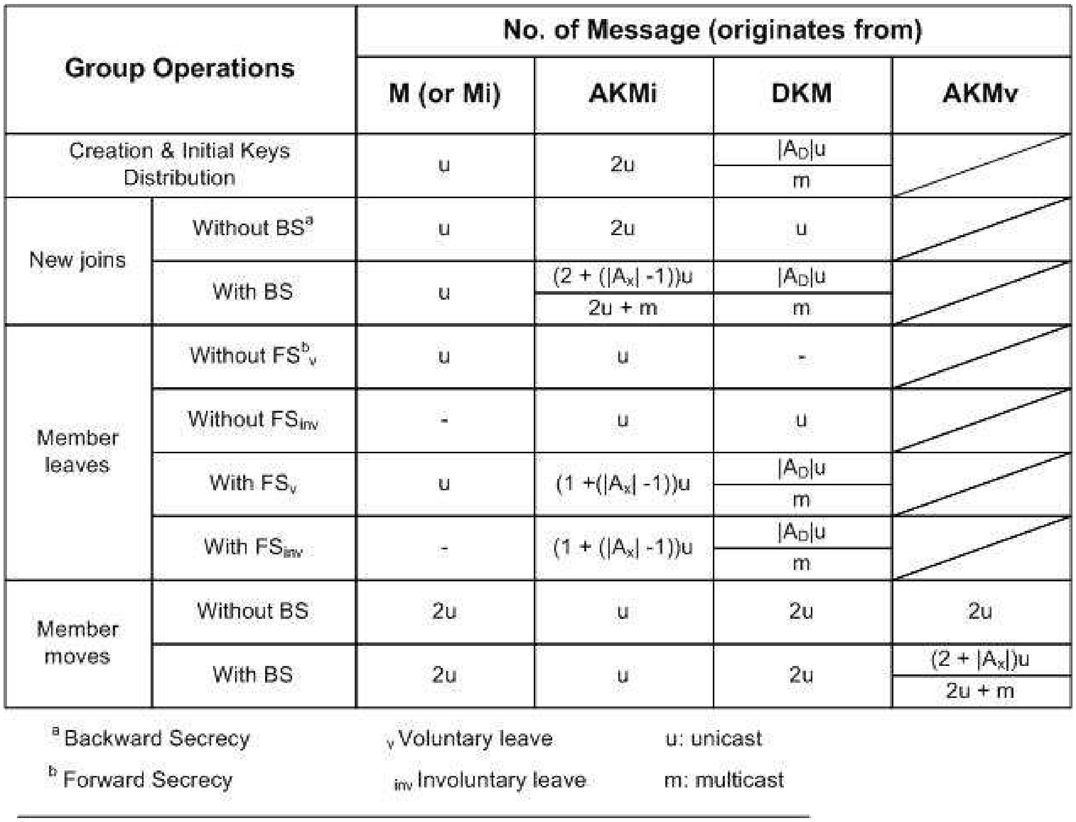
Table 10.5: Cost assessment based on storage complexity.

We conclude that the main cost of key storage is reasonably distributed amongst key managers (DKM and AKMs), while keeping the cost of key storage at group members M minimal. A group member with a typical mobile device 206MHz processor with 64MB of RAM (as mentioned earlier) can comfortably cope with the total cost of 3+$|h_{Mob}|$ keys storage (see *Table 10.5*), with each key length of 128 bits (as discussed in *Section 8.3.6*). Also, as the main key manager in a domain, DKM usually carries a lot of weight as the primary entity for managing group operations. The load for storing keys is shared with other key managers (AKMs) in a domain, hence the operational load is reasonably balanced amongst DKM and AKMs. For example, DKM does not need to keep *Area-Member* key pairs shared between an AKM and a group member, which are managed at the area level by the AKM.

We observe that the number of keys kept by DKM increases as the number of multicast groups increases. Similarly, the number of keys kept by an AKM increases as the number of group members residing in that area increases.

(d) *Communication complexity*

This assessment demonstrates the framework performance with respect



Group Operations		No. of Message (originates from)			
		M (or Mi)	AKMi	DKM	AKMv
Creation & Initial Keys Distribution		u	2u	$\lvert A_D \rvert u$ m	
New joins	Without BS[a]	u	2u	u	
	With BS	u	$(2 + (\lvert A_x \rvert -1))u$ $2u + m$	$\lvert A_D \rvert u$ m	
Member leaves	Without FS[b]$_v$	u	u	-	
	Without FS$_{inv}$	-	u	u	
	With FS$_v$	u	$(1 +(\lvert A_x \rvert -1))u$	$\lvert A_D \rvert u$ m	
	With FS$_{inv}$	-	$(1 + (\lvert A_x \rvert -1))u$	$\lvert A_D \rvert u$ m	
Member moves	Without BS	2u	u	2u	2u
	With BS	2u	u	2u	$(2 + \lvert A_x \rvert)u$ $2u + m$

[a] Backward Secrecy $\quad$ $_v$ Voluntary leave $\quad$ u: unicast

[b] Forward Secrecy $\quad$ $_{inv}$ Involuntary leave $\quad$ m: multicast

Table 10.6: Cost assessment based on communication complexity.

to the number of messages sent by each communicating entity (key managers and group member) involved in group operations. For every group operation, one cost is incurred when:

- A *unicast* message is sent, and we denote this by a u.
- A *multicast* message is sent, and we denote this by an m.

This is summarized in *Table 10.6.* Note that we do not specify where the messages were being sent to, but rather analyze the number of messages originating from a particular entity.

From *Table 10.6*, we observe that the number of messages sent by a group member M throughout group operations is reasonably low, at most $2u$. The cost incurred during new join and leave operations (with provision of backward and forward secrecy) varies depending on whether a *unicast* or *multicast* message is sent.

For example, in new joins with backward secrecy (see *2nd column:* AKM_i), the cost from AKM$_i$ is $(2+(|A_x|-1))u$, which is equivalent to *two unicast* messages plus $|A_x|-1$ *unicast* messages, which were sent to group members in an area (excluding the newly joined member). This cost from AKM$_i$ can be reduced significantly if *multicast* is used. In the same example, it is reduced to $2u+m$, which is a total cost of just three messages.

Similarly, DKM can reduce the cost of messages sent to all AKMs in the domain (see *3rd column: DKM*) by using *multicast*, which costs only *one* message, instead of $|A_D|$ messages if *unicast* is used.

By using the multicast functionality, a message intended to a group of recipients, such as all group members in an area, can be sent once by the AKM of that area. This is important in WMobEs where only limited bandwidth is available. A typical wireless network with 1700-1800 MHz (for upload and download traffic) (IIyas, 2003) connected using a wireless access point at 115Kbps data rate (Potlapally et al., 2006), can still comfortably cope with the number of messages exchanged during the group operations. As discussed in *Section 8.2.3*, we can conclude that the communication overhead and the bandwidth usage are suitably low for the intended environments.

10.4 Summary

In this chapter we have assessed the proposed framework. We have shown the extent to which the framework meets its specified requirements and design objectives. These, we believe have been addressed and achieved reasonably well, although the actual feasibility of the framework can probably only be verified through practical implementation.

Chapter 11

Conclusions and Future Work

This chapter concludes our work and provides directions for future research.

In *Section 11.1* we present the research achievements of the thesis. Finally, in *Section 11.3*, we provide several suggestions for future work.

11.1 Research Achievements

Multicast functionality enables group communication to occur between groups of network hosts over vast and open networks. There are many applications, such as video conferencing and digital content broadcasting, that require communication between groups of entities. Such applications have a strong need for communication security, which is conventionally supplied through the implementation of appropriate cryptographic mechanisms, whose security in turn is provided by managing the cryptographic keys involved.

With the increasing popularity and demand for group-based applications, the need for security has become more important. This effort has been recognized by the network community who included multicast functionality as part of the IPv6 (Internet protocol version 6) design for enabling multicast group communication.

As most proposals are intended for deployment in wired environments, little consideration has been given for deployment in other networking environments. Applications which were made available in wired environments should also be made available in wireless environments. With the increasing need for host mobility in networked devices, it is thus important to develop security techniques for group communication in mobile environments.

The overall goal of this thesis is to specify a group key management framework for secure group communication in wireless mobile environments.

We started our investigation by looking at the main security (research) problem areas in multicast group communication, and we identified that the main issues pertaining to multicast group communication security are group membership, key management processes (protocols), group security and authentication, and scalability. Each of these affects how cryptographic keys are managed.

While many key management issues are generic to any networking environment, we have identified specific issues necessary for establishing secure group communication in wireless mobile environments. To provide us with a blueprint for a detailed specification of a GKMF in such environments, we proposed a generic model of a suitable GKMF.

Based on the generic model, we then described a specification of a GKMF for secure group communication, based on a specific wireless mobile architecture with a fixed infrastructure that supported key management entities for the provision of key management services. Our proposed GKMF has a number of notable features:

(a) It draws on several design properties of previous GKMFs that facilitate scalable key management. In particular, the architecture is based on the notion of domains and areas to contain the impact of re-keying events.

(b) It includes a protocol to support members moving from one key management area to another. This facility has not been provided in previous GKMFs, and is one of several features that are incorporated explicitly for deployment in wireless mobile environments.

(c) To support membership changes and host mobility, we introduce the use of lists. The *HisList* allows a key manager to keep track of member leaving a multicast group and the reason for the leave. This is useful when the same member wishes to re-join the group at a later time. The *MobList* allows key managers in a domain to manage host mobility more efficiently. This list can be used to keep track of host mobility, so that when a member moves into an area, a key manager can determine whether the member is a returning member or a first time mover into the area. In the case where a member is moving back into a previously visited area, re-keying may not always be necessary.

We have conducted a basic analysis of the proposed GKMF to assess the extent to which the framework meets its specified objectives and design requirements. This analysis was conducted at the level of the framework as a whole, as well as at the level of specific protocols. The analysis considered general issues, security issues and performance issues. We have argued that in general our GKMF meets the identified requirements for implementation in wireless mobile environments.

To conclude, we believe that this thesis has achieved its research aim and objectives as stated in *Chapter 1*, as follows:

- We have provided adequate backgrounds on multicast technology and its capability to enable group communication.
- We have identified different security challenges for establishing secure group communication in wired and wireless networks.
- We have established a generic GKMF, which describes the essential and desirable components that need to be addressed if a GKMF for WMobEs is to be specified.
- We have designed a GKMF for secure group communication for a WMobE (with fixed infrastructure). This includes specifications of the main components and key management protocols identified within the GKMF.

- We have conducted basic analysis of the proposed GKMF in terms of functionality, security and performance to see the extent that the framework meets its design requirements and its security objectives.

11.2 Research Limitations

We have demonstrated that this thesis has adequately achieved its research aim and objectives for establishing secure group communication in a wireless mobile environment.

In this section, we list out research limitations of this thesis, which may be useful for researching future work (see also *Section 11.3*):

(a) We focused our GKMF for provision of backward and/or forward secrecy (confidentiality services), which primarily determined by appropriate key management techniques for updating cryptographic keys. We thus addressed simplified protocols where provision of other security services such as entity authentication and data origin authentication is implicitly assumed. We do not explicitly treat these here because their solutions can be provided using generic techniques, which are not specific to multicast group communication.

(b) One of the main security research issues that we attempted is scalability. From the perspective of key management for group communication, scalability primarily affects the methods by which cryptographic keys are updated. Host mobility in dynamic wireless mobile environments may aggravate the scalability issue further as group members are not only allowed to join and/or leave a multicast group at any time (for dynamic group membership), also allowed to move between areas while still remaining in a group session. Our GKMF proposed to place group members (each member may belong to different multicast groups) in an area, in order to alleviate scalability problem which may occur in particular for re-keying the traffic key during group operations (such as during member joining and leaving protocols, see *Chapter 9*).

As group members of a multicast group may disperse across multiple areas, re-keying of the traffic key (which is unique to a multicast group) may have to span all areas in a domain, which could result in scalability problem because all areas will have to process the re-keying whenever there is a change in group membership (for backward and/or forward secrecy).

This problem may be mitigated by placing group members of a multicast logically in an area. However, in practice this may not be the case, as group members may physically exist and move between areas. Every member (or every area where the member residing) may still need to be treated and processed individually.

(c) The result on the performance of framework is done based on paper analysis. Although the actual feasibility of framework can probably only be verified through practical implementation, results obtained from the analysis may be useful prior to the implementation.

11.3 Future Work

While this thesis provides a useful contribution to the understanding of group key management in wireless mobile environments, there is plenty of scope for further investigation of this subject.

There are two different directions in which the work covered by this thesis could be extended.

(1) Further analysis of the proposed GKMF. We have only provided a partial specification and analysis of the proposed GKMF. This could be extended by:

- Fully specifying the GKMF for a particular type of wireless mobile environment, down to protocol level.
- Adding extra key management functionality, such as support for key recovery.

- Conducting a more formal security analysis of the fully specified protocols.
- Implementing the GKMF and conducting a more detailed performance analysis.

While we have conducted an acceptable paper analysis of our GKMF, the above further research activities would provide the ultimate test of the proposal's viability. This work requires a greater time resource than was available in the study period covered by this thesis, but provides a natural next step.

(2) This thesis includes a generic GKMF for establishing secure group communication in wireless mobile environments. It can thus be used as a basis for developing other GKMFs for different types of wireless mobile environments. Particular design changes that would lead to alternative GKMF designs are:

- *Other WMobEs.* To extend the framework design to other wireless mobile networks such as wireless MANETs.
- *Asymmetric key cryptography.* To use the generic framework to design a GKMF for WMobEs where the use of asymmetric cryptography was deemed acceptable from a performance perspective.

Bibliography

Almeroth, K. C. (2000). The Evolution of Multicast: From the MBone to Inter-Domain Multicast to Internet2 Deployment. *Network IEEE*, 14(1):10–20.

Ammer, C. (2000). *The American Heritage Dictionary of the English Language, Fourth Edition*. Houghton Mifflin Company.

Apple (2007). iPhone: Internet in Your Pocket, published by Apple Inc. http://www.apple.com/iphone/.

Ballardie, A. (1996). *Scalable multicast key distribution*. RFC 1949.

Baugher, M., Canetti, R., Dondeti, L., and Lindholm, F. (2003). *Group Key Management Architecture*. Internet Draft IETF MSEC WG. http://www2.tools.ietf.org/html/draft-ietf-msec-gkmarch-04.

Baugher, M., Canetti, R., Dondeti, L., and Lindholm, F. (2005). *Multicast Security (MSEC) Group Key Management Architecture*. RFC 4046.

BBC (2007). Launch date for iPhone revealed, reported by BBC News. http://news.bbc.co.uk/1/hi/technology/6717865.stm.

Bhargava, B., Kamisety, S. B., and Madria, S. K. (2000). Fault-tolerant authentication and group key management in mobile computing. Technical report, Center for Education and Research in Information Assurance and Security, and Department of Computer Science Purdue University. http://www.cs.purdue.edu/homes/bb/cs690b/report.ps.

Bruschi, D. and Rosti, E. (2002). Secure multicast in wireless networks of mobile hosts: Protocols and issues. *Mobile Networks and Applications*, 7(6):503–511.

BS (1997). *Information technology - Security techniques - Entity authentication - Part 1: General (BS ISO/IEC 9798-1)*. British Standards.

BS (2002). *Information technology - Security techniques - Time-stamping services - Part 1 (BS ISO/IEC 18014-1)*. British Standards.

Canetti, R., Garay, J., Itkis, G., Micciancio, D., Naor, M., and Pinkas, B. (1999). Multicast security: A taxonomy and some efficient constructions. In *Proceeding of IEEE Eighteenth Annual Joint Conference of the IEEE Computer and Communications Societies (INFOCOMM)'99*. http://citeseer.ist.psu.edu/canetti99multicast.html.

Caronni, G., Lubich, H., Aziz, A., Markson, T., and Skrenta, R. (1996). SKIP: Securing the internet. In *Proceedings of WET ICE '96 Fifth Workshop on Enabling Technologies*, pages 62–67. http://citeseer.ist.psu.edu/caronni96skip.html.

Casner, S. and Deering, S. (1992). First IETF Internet Audiocast. *SIGCOMM Computer Commununication Review*, 22(3):92–97. http://citeseer.ist.psu.edu/casner92first.html.

Chlamtac, I. and Redi, J. (1998). *Mobile Computing: Challenges and Potential*. Encyclopedia of Computer Science, 4th Edition, International Thomson Publishing.

Ciscosystem (2006). Cisco internetworking terms and acronyms. http://www.cisco.com/univercd/cc/td/doc/cisintwk/ita/index.htm.

Comer, D. E. (2001). *Computer Networks and Internets with Internet Applications*, pages 99–137. Prentice Hall, third edition.

Dankers, J., Garefalakis, T., Schaffelhofer, R., and Wright, T. (2004). PKI in mobile systems. In *Security for Mobility by C. J. Mitchell*, pages 11–32. The Institution of Electrical Engineers (IEEE), London UK.

Davies, J. (2003). *Understanding IPv6*. Microsoft Press.

Decleene, B., Dondeti, L., Griffin, S., Hardjono, T., Kiwior, D., Kurose, J., Towsley, D., Vasudevan, S., and Zhang, C. (2001). Secure group communications for wireless networks. In *Proceedings of IEEE MILCOM'01*, pages 66–73.

Deering, S. (1989). *Host extensions for IP multicasting*. RFC 1112.

Devereaux-Weber, D. (2006). Mbone (internet multicasting backbone) and multimedia resources. http://www.mbone.net/, owned by Solution Box, Inc.

Diot, C., Levine, B. N., Lyles, B., Kassem, H., and Balensiefen, D. (2000). Deployment issues for the ip multicast service and architecture. *Network IEEE*, 14(1):78–88.

FIPS (2001). *Advanced Encryption Standard (AES)*. National Institute of Standards & Technology (NIST). Federal Information Processing Standards Publication 197 (FIPS PUB 197), http://csrc.nist.gov/publications/fips/fips197/fips-197.pdf.

FIPS (2004). *Standards for Security Categorization of Federal Information and Information Systems*. National Institute of Standards & Technology (NIST). Federal Information Processing Standards Publication 199 (FIPS PUB 199), http://csrc.nist.gov/publications/fips/fips199/FIPS-PUB-199-final.pdf.

Forman, G. H. and Zahorjan, J. (1994). The challenges of mobile computing. Technical report, Computer Science & Engineering, University of Washington. TR-93-11-03, http://citeseer.ist.psu.edu/article/forman94challenges.html.

Gong, L. and Shacham, N. (1995). Multicast security and its extension to a mobile environment. *Wireless Networks*, 1(3):281–295.

Gove, R. A. (2000). Fundamentals of cryptography and encryption. In *Information Security Management Handbook, 4th Edition by H. F. Tipton and M. Krause*. Auerbach.

Goyeneche, J.-M. D. (2004). Multicast over TCP/IP HOWTO. http://www.tldp.org/HOWTO/Multicast-HOWTO.html.

GSEC (2007). The Group Security Research Group (GSEC) of Internet Research Task Force (IRTF). http://www.securemulticast.org/gsec-index.htm.

Hardjono, T., Cain, B., and Doraswamy, N. (2000a). *A Framework for Group Key Management for Multicast Security*. Internet Draft IETF. http://www3.ietf.org/proceedings/00jul/I-D/ipsec-gkmframework-02.txt.

Hardjono, T., Cain, B., and Monga, I. (2000b). *Intra-Domain Group Key Management Protocol.* Internet Draft IETF. http://www.securemulticast.org/draft-ietf-ipsec-intragkm-03.txt.

Hardjono, T. and Dondeti, L. R. (2003). *Multicast and Group Security.* Artech House.

Hardjono, T. and Tsudik, G. (2000). IP Multicast Security: Issues and Directions. *Annales de Telecom*, pages 324–340. http://citeseer.ist.psu.edu/hardjono99ip.html.

Hardjono, T. and Weis, B. (2004). *The Multicast Group Security Architecture.* RFC 3740.

Harkins, D. and Carrel, D. (1998). *The Internet Key Exchange (IKE).* RFC 2409.

Harney, H. and Muckenhirn, C. (1997). *Group Key Management Protocol (GKMP) specification.* RFC 2093.

Hillebrand, F. (2002). *GSM and UMTS: The Creation of Global Mobile Communication.* John Wiley & Sons, Ltd.

Hinden, R. and Deering, S. (2006). *IP Version 6 Addressing Architecture.* RFC 4291.

Huitema, C. (1995). *Routing in the Internet.* Prentice Hall.

IANA (2005). Internet Protocol v4 Multicast Address Assignments, IP Version 6 Addressing Architecture. Internet Assigned Numbers Authority (IANA) (Standard Documents), http://www.iana.org/ipaddress/ip-addresses.htm.

IETF (2007). The Internet Engineering Task Force (IETF). http://www.ietf.org/home.html.

IIyas, M. (2003). *The Handbook of Ad Hoc Wireless Networks.* CRC Press.

Ikbal, J. (2003). An introduction to cryptography. In *Information Security Management Handbook, 4th Edition by H. F. Tipton and M. Krause.* Auerbach.

IRTF (2007). Internet Research Task Force (IRTF). http://www.irtf.org/.

ISO (1988). *Information technology-Banking-Key Management (ISO/IEC 8732).* International Standard.

ISO (1989). *Information processing systems - Open Systems Interconnection - Basic Reference Model - Part 2: Security Architecture (ISO 7498-2)*. International Standard.

ISO (1994a). *Information technology - Security techniques - Data integrity mechanism based on H-MAC algorithm (ISO/IEC 9797-2)*. International Standard.

ISO (1994b). *Information technology - Security techniques - Data integrity mechanism using a cryptographic check function employing a block cipher algorithm (ISO/IEC 9797-1)*. International Standard.

ISO (1996a). *Information technology - Security techniques - Key management - Part 1: Framework (ISO/IEC 11770-1)*. International Standard.

ISO (1996b). *Information technology - Security techniques - Key management - Part 2: Mechanism using symmetric techniques (ISO/IEC 11770-2)*. International Standard.

ISO (1999a). *Information technology - Security techniques - Entity authentication - Part 2: Mechanisms using symmetric encipherment algorithms (ISO/IEC 9798-2)*. International Standard.

ISO (1999b). *Information technology - Security techniques - Entity authentication - Part 4: Mechanisms using a cryptographic check function (ISO/IEC 9798-4)*. International Standard.

Lin, Y. and Chlamtac, I. (2001). *Wireless and Mobile Network Architectures*. John Wiley & Sons, Inc.

MatKiah, M. L. and Martin, K. M. (2005). Group communication: Design challenges in the development of key management frameworks in wireless mobile environments. In *Proceedings of International Conference on Security and Management SAM'05*, pages 385–390. CSREA Press.

MatKiah, M. L. and Martin, K. M. (2006). A generic group key management framework for group communication in wireless mobile environments. In *Proceedings of the Sixth International Network Conference INC2006*, pages 347–354. University of Plymouth.

Maughan, D., Schertler, M., Schneider, M., and Turner, J. (1998). *Internet security association and key management protocol (ISAKMP)*. RFC 2408. http://www.ietf.org/rfc/rfc2408.txt.

McDaniel, P., Prakash, A., and Honeyman, P. (1999). Antigone: A flexible framework for secure group communication. In *Proceedings of the 8th USENIX Security Symposium*, pages 99–114. USENIX. Washington D.C., USA, August 23-26.

Michiardi, P. and Molva, R. (2006). Ad hoc network security. In *Handbook of Information Security: Key Concepts, Infrastructure, Standards, and Protocols, Volume 1 Editor-in-chief H. Bidgoli*. John Wiley & Sons, Inc.

Miller, C. K. (1999). *Multicast Networking and Applications*. Addison Wesley.

Mittra, S. (1997). Iolus: A framework for scalable secure multicasting. In *Proceedings of ACM SIGCOMM*, pages 277–288, Cannes, France.

MSEC (2007). Multicast Security (MSEC) Group of Internet Engineering Task Force (IETF). http://www.ietf.org/html.charters/msec-charter.html.

Murray, W. H. (2000). Principles and applications of cryptographic key management. In *Information Security Management Handbook, 4th Edition by H. F. Tipton and M. Krause*. Auerbach.

Nichols, R. K. and Lekkas, P. C. (2002). *Wireless Security: Models, Threats, and Solutions*. McGraw-Hill.

Noubir, G., Zhu, F., and Chan, A. H. (2002). Key management for simultaneous join/leave in secure multicast. In *Proceedings of IEEE International Symposium on Information Theory*, pages 325–331. http://citeseer.ist.psu.edu/552621.html.

Park, J., Suh, Y., and Kang, S. (2002). Supporting mobile multicast in mobile networks by considering host mobility. In *IDMS/PROMS 2002: Proceedings of the Joint International Workshops on Interactive Distributed Multimedia Systems and Protocols for Multimedia Systems*, pages 263–273. Springer-Verlag.

Perrig, A. and Tygar, J. (2003). *Secure Broadcast Communication in Wired and Wireless Networks*. Kluwer Academic Publishers.

Pessi, P. (2003). Secure multicast. http://www.tml.tkk.fi/Opinnot/Tik-110.501/1995/multicast.html.

Potlapally, N. R., Ravi, S., Raghunathan, A., and Jha, N. K. (2006). A study of the energy consumption characteristics of cryptographic algorithms and security protocols. *Proceedings of IEEE Transactions on Mobile Computing*, 5(2):128–143.

Reid, B. (1997). What is multicast? Edited archived of the SCADA mailing list, http://members.iinet.net.au/ ianw/archive/x1584.htm.

Rodeh, O., Birman, K., and Dolev, D. (2000). Optimized group rekey for group communication systems. In *Proceedings of ISOC Network and Distributed Systems Security*, pages 39–48, San Diego, CA. Cornell University.

Savetz, K., Randall, N., and Lepage, Y. (1998). *MBONE:Multicasting Tomorrow's Internet*. http://www.savetz.com/mbone/, Printed copy available from John Wiley & Sons Inc.

Setia, S., Koussih, S., Jajodia, S., and Harder, E. (2000). Kronos: A scalable group re-keying approach for secure multicast. In *Proceedings of IEEE Symposium on Security and Privacy*, pages 215–228. IEEE Computer Society.

Setia, S., Zhu, S., and Jajodia, S. (2002). A scalable and reliable key distribution protocol for multicast group rekeying. Technical report, Center for Secure Information Systems, George Mason University. http://citeseer.ist.psu.edu/setia02scalable.html.

SMuG (2007). Secure Multicast Research Group (SMuG) of Internet Research Task Force (IRTF). http://www.irtf.org/old-groups.

Stallings, W. (1999). *Cryptography and Network Security: Principles and Practice*. Prentice Hall.

Steiner, M., Tsudik, G., and Waidner, M. (1998). CLIQUES: A new approach to group key agreement. In *Proceedings of the 18th International Conference on Distributed Computing Systems (ICDCS'98)*, pages 380–387, Amsterdam. IEEE Computer Society Press.

TechNews (2007). UK Survey: 7 million keen to buy iPhone. http://www.tech.co.uk/gadgets/phones/mobile-phones/news/uk-survey-7-million-keen-to-buy-iphone?articleid=540737775.

Vines, R. D. (2002). *Wireless Security Essentials: Defending Mobile Systems from Data Piracy*. Wiley.

Vodafone (2007). About Vodafone UK. http://online.vodafone.co.uk.

Wadaa, A., Olariu, S., Wilson, L., and Eltoweissy, M. (2004). Scalable cryptographic key management in wireless sensor networks. In *ICDCSW '04: Proceedings of the 24th International Conference on Distributed Computing Systems Workshops - W7: EC (ICDCSW'04)*, pages 796–802. IEEE Computer Society Press.

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*, 17(8).

Wallner, D., Harder, E., and Agee, R. (1999). *Key Management for Multicast: Issues and Architectures*. RFC 2627.

Williamson, B. (2000). *Developing IP Multicast Networks*. Cisco Press. Chapter 43.

Wittmann, R. and Zitterbart, M. (2001). *Multicast Communication: Protocols and Applications*. Morgan Kaufmann.

Wong, C. K., Gouda, M. G., and Lam, S. S. (1998). Secure group communications using key graphs. In *Proceedings of the ACM SIGCOMM '98 conference on Applications, technologies, architectures, and protocols for computer communication*, pages 68–79. ACM Press. http://citeseer.ist.psu.edu/article/wong98secure.html.

Zhang, C., Decleene, B., Kurose, J., and Towsley, D. (2002). Comparison of inter-area rekeying algorithms for secure wireless group communications. *An International Journal Performance Evaluation*, 49:1–20.

Zou, X. and Thukral, A. (2006). Key management. In *Handbook of Information Security: Information Warfare; Social, Legal, and International Issues; and Security Foundations, Volume 2 Editor-in-chief H. Bidgoli*. John Wiley & Sons, Inc.