# A LIGHTWEIGHT KEY MANAGEMENT FRAMEWORK FOR SECURE DYNAMIC GROUP BASED APPLICATIONS

**SALMAN IQBAL MALIK**

**FACULTY OF COMPUTER SCIENCE AND INFORMATION TECHNOLOGY**
**UNIVERSITY OF MALAYA**
**KUALA LUMPUR**

**2017**

# A LIGHTWEIGHT KEY MANAGEMENT FRAMEWORK FOR SECURE DYNAMIC GROUP BASED APPLICATIONS

**SALMAN IQBAL MALIK**

**THESIS SUBMITTED IN FULFILMENT OF THE REQUIREMENTS FOR THE DEGREE OF DOCTOR OF PHILOSOPHY**

**FACULTY OF COMPUTER SCIENCE AND INFORMATION TECHNOLOGY**
**UNIVERSITY OF MALAYA**
**KUALA LUMPUR**

**2017**

**UNIVERSITY OF MALAYA**

**ORIGINAL LITERARY WORK DECLARATION**

Name of Candidate: Salman Iqbal (I.C/Passport No: )

Matric No:

Name of Degree: Doctor of Philosophy

Title of Project Paper/Research Report/Dissertation/Thesis ("this Work"): A LIGHTWEIGHT KEY MANAGEMENT FRAMEWORK FOR SECURE DYNAMIC GROUP BASED APPLICATIONS

Field of Study: Computer Science (Network Security)

I do solemnly and sincerely declare that:

(1) I am the sole author/writer of this Work;
(2) This Work is original;
(3) Any use of any work in which copyright exists was done by way of fair dealing and for permitted purposes and any excerpt or extract from, or reference to or reproduction of any copyright work has been disclosed expressly and sufficiently and the title of the Work and its authorship have been acknowledged in this Work;
(4) I do not have any actual knowledge nor do I ought reasonably to know that the making of this work constitutes an infringement of any copyright work;
(5) I hereby assign all and every rights in the copyright to this Work to the University of Malaya ("UM"), who henceforth shall be owner of the copyright in this Work and that any reproduction or use in any form or by any means whatsoever is prohibited without the written consent of UM having been first had and obtained;
(6) I am fully aware that if in the course of making this Work I have infringed any copyright whether intentionally or otherwise, I may be subject to legal action or any other action as may be determined by UM.

Candidate's Signature Date:

Subscribed and solemnly declared before,

Witness's Signature Date:

Name:

Designation:

## ABSTRACT

In recent years, group based applications have gained popularity due to their interesting and promising functionalities such as video on demand, teleconferencing and pay per view. The development of wireless networks and the emergence of mobile devices such as smartphones and tablets have also increased the demands for group based applications. These applications facilitate real-time information exchange among a large number of users over a wireless mobile environment. Due to the open nature of the wireless mobile environment, these applications are vulnerable to various kinds of security threats and thus, security is a major concern. Group Key Management Protocols (GKMPs) provide a secure way of communicating with group members across multiple entities. GKMPs provide member authentication, access control and management of the keying material. However, the implementation of GKMPs leads to significant computational, storage and communication overheads as well as potential system bottlenecks due to the high mobility of group members. Thus, the major issue that needs to be addressed in GKM is to minimize the computational, storage and communication overheads. The goal of this research is to address these issues and design a lightweight key management framework which requires fewer computations of keys for dynamic mobile users. A new group key management framework is proposed in this research, which is called the "DynaMic Group Key Management" (DM-GKM) framework. This framework exploits the advantages of an asymmetric key cryptosystem in order to guarantee security and it alleviates the rekeying overhead and distributing the independent Group Key (GK) for each cluster. Simulation and performance analysis demonstrate that the DM-GKM framework fulfils the requirements of a lightweight key management framework for large, dynamic groups of users. An analytical model, formal analysis and statistical analysis are also developed to determine the performance and security features of the proposed framework.

## ABSTRAK

Dalam tahun-tahun kebelakangan ini, aplikasi berasaskan kumpulan telah mendapat populariti kerana fungsi menarik yang dijanjikan seperti video atas permintaan, telekonferensi dan bayar untuk setiap pandangan. Perkembangan rangkaian tanpa wayar dan kemunculan peranti mudah alih seperti telefon pintar dan tablet juga telah meningkatkan permintaan untuk aplikasi berasaskan kumpulan. Aplikasi ini memudahkan pertukaran maklumat masa nyata di kalangan sebilangan besar pengguna melalui persekitaran mudah alih tanpa wayar. Disebabkan sifat terbuka persekitaran mudah alih tanpa wayar, aplikasi ini terdedah kepada pelbagai jenis ancaman keselamatan dan oleh itu, keselamatan menjadi kebimbangan utama. Protokol Pengurusan Kunci Kumpulan (GKMPs) menyediakan cara selamat untuk berkomunikasi dengan ahli kumpulan merentas pelbagai entiti. GKMP menyediakan pengesahan ahli, kawalan akses dan pengurusan bahan utama. Walau bagaimanapun, pelaksanaan GKMP membawa kepada lebihan yang besar, penyimpanan dan komunikasi yang signifikan serta berpotensi untuk kemusnahan sistem disebabkan oleh mobiliti tinggi anggota kumpulan. Oleh itu, isu utama yang perlu ditangani dalam GKM adalah meminimumkan lebihan yang besar, penyimpanan dan komunikasi. Matlamat penyelidikan ini adalah untuk menangani isu-isu ini dan mereka bentuk rangka kerja pengurusan utama yang ringan yang memerlukan pengiraan kekunci yang lebih sedikit bagi pengguna mudah alih dinamik. Rangka kerja pengurusan kunci kumpulan baru dicadangkan dalam penyelidikan ini, yang dikenali sebagai rangka kerja "Pengurusan Kunci Kumpulan DynaMic" (DM-GKM). Rangka kerja ini memanfaatkan kelebihan cryptosystem utama asimetri untuk menjamin keselamatan dan mengurangkan kelebihan input semula dan mengagihkan Kelompok Kunci Bebas (GK) untuk setiap kelompok. Analisis simulasi dan prestasi menunjukkan bahawa rangka kerja DM-GKM memenuhi keperluan rangka kerja pengurusan kunci yang ringan untuk kumpulan

pengguna yang besar dan dinamik. Model analisis, analisis formal dan analisis statistik juga dibangunkan untuk menentukan ciri prestasi dan keselamatan rangka kerja yang dicadangkan.

## ACKNOWLEDGEMENT

All praises to Allah for the strengths and His blessing in completing this thesis.

Foremost, I would like to thanks to my parents and family. Without their continuous support and encouragement I never would have been able to achieve my goals.

I would like to express my sincere gratitude to my supervisor Prof. Dr. Miss Laiha Mat Kiah for the continuous support of my Ph.D study and research, for her motivation, enthusiasm, and immense knowledge. Her guidance helped me in all the time of research and writing of this thesis.

I would also like to express my appreciation to my co-supervisor Associate Prof. Dr. Nor Badrul Anuar Bin Juma'at for his support and help towards my PhD completion.

Last, but not least, I would like to thank my fellow lab mates and friends, especially Dr. Babak Daghighi, Dr. Suleman Khan, Dr. Muhamad Habib Ur Rahman, Ahmad Firdaus Zainal Abidin, and Faizal Razak for their feedback, cooperation and friendship.

## Table of Contents

## LIST OF FIGURES

## LIST OF TABLES

## LIST OF ABBREVIATIONS

| | |
|---|---|
| SGC | Secure Group Communication |
| DM-GKM | Dynamic Group Key Management |
| $G^{TEK}$ | Group Traffic Encryption Key |
| $G^{TEKi}$ | Independent TEK for each subgroup. |
| $S^{KEK}$ | Sub-Group Key |
| GKM | Group Key Management |
| GKMF | Group Key Management Framework |
| AGKS | Area Group Key Server |
| TCP | Transmission Control Protocol |
| UDP | User Datagram Protocol |
| DGKS | Domain Group Key Server |
| TEK | Traffic Encryption Key |
| KEK | Key Encryption Key |
| CRT | Chinese Remainder Theorem |
| RSA | Ron Rivest, Adi Shamir, Leonard Adleman |
| $C_m$List | Current member list |
| GKM | Group Key Management |
| $O_m$List | Old member list |
| $M_m$List | Member's mobility list |
| SG | Subgroup |
| CO | Communication Overhead |
| IoT | Internet of Things |
| DKD | Domain Key Distributor |

| | |
|---|---|
| AKD | Area Key Distributors |
| SO | Storage Overhead |

# CHAPTER 1: INTRODUCTION

Over the last few decades, point-to-point communication such as the User Datagram Protocol (UDP) and Transmission Control Protocol (TCP) has been a major part of communication networks. These models are generally designed for applications that require no more than two processes at a time. Such models use Internet Protocol (IP) routing techniques at a low level over a network such as the Internet Group Management Protocol (IGMP) (Fenner, 1997). However, even though these technologies fulfil their purpose, they do not fulfil the desired success metrics due to the incompatibility of routers, implementation cost and lack of support from Internet providers.

In the last few years, there is explosive advancement of networking and information technologies such as online conferencing, online gaming, military communication and Internet Protocol television (IPTV), which inspires the development of group communication. Group communication allows a host to send data simultaneously to a group of other hosts. This phenomenon prevents the establishment of point-to-point connections among the group. Nowadays, application-level group communication has superseded point-to-point communication since it provides the same functionality at a lower cost with minimal deployment efforts. Application-level group communication becomes easier with the advent of wireless mobile technologies which provide efficient installation and compatibility between different services. According to statistics, mobile devices represent 65% of all digital media, and mobile connected devices per capita will reach 1.5 by 2020 (Index, 2013). This clearly shows the popularity of emerging mobile devices and the growth of wireless technology users. The escalating trend in mobile device usage increases the demand for group based applications such as video conferencing, online games and e-health systems. Group based applications enable users

to send data to multiple receivers simultaneously, whereby a single message is sent to a selected group of recipients. These applications provide essential services that simplify real-time information exchange among a large number of mobile users. The communication cost is reduced significantly since the data is transmitted to a group of users simultaneously. Group based applications can be either static or dynamic. As the name implies, the membership of the group in static group based applications is closed, pre-determined and invariant during the group communication session. An example of a static group based application is a short conference meeting between the members of a project team in order to clarify certain aspects of the project. However, these applications may suffer from useless computational overhead. In dynamic group based applications, the membership changes during the group communication session. However, designing an efficient and scalable dynamic group based application is challenging due to the frequency of the members joining and leaving the group as well as the mobility of the members. In the next generation of wireless mobile communications, group based applications are essential to support real-time communication between large groups of users.

In general, group based applications require a secure communication channel to prevent disclosure of information to unauthorized users. However, group based communications make use of an open wireless network which is vulnerable to several attacks, resulting in an insecure communication environment. For example, security is of utmost importance in private conferences due to the provision of data confidentiality in a dynamic group membership. This means that only the authorized group members can properly access the data. For this reason, it is necessary to have an efficient key management mechanism, which can efficiently manage a large, dynamic group of mobile users.

Key management methods are used for secure key generation and immediate revocation of keys whenever there is a change in the membership. GKs are used for security and privacy reasons, whereby the GKs are shared among users of the group as an access control mechanism. A message is encrypted only once for the group and the message is then transmitted to the group and further decrypted using the GK. The data traffic is encrypted with a single key and this key is then distributed to the group using one of the several mechanisms. The GKMP serves to generate, update and distribute the GK and user private keys within the group in a secure manner (Gharout et al., 2010; Trust Tshepo Mapoka, 2013). The fundamental principle in key management architecture is to obtain a valid GK in order to authorize an entity and participate in the operation of communication by using the key. The group key is used to send data to all authorized members within the group. Thus, the basic task of GKM is to distribute the group key so that the group members having the valid key can recover the original message. This prevents access to unauthorized members. In other words, this key is able to prevent other users from accessing the content of the group. Hence, it is crucial to have a robust mechanism to distribute the group key for dynamically changing group members in a wireless mobile environment (Cao, Liao, & Wang, 2006; Kiah & Martin, 2007).

In group based applications, the group members are dynamically changing because they require multiple key updates due to frequent membership changes during the group communication session. In addition, the group key must be changed when the duration of an authorized member has expired or when a member leaves the group. In this case, it is necessary to generate and distribute a new key so that the leaving group member is no longer able to access the content of other members. Hence, the sender of the group needs to share the new group key to all legitimate group members with the exception of

the leaving member. This phenomenon is known as rekeying. Rekeying should be carried out whenever an old member leaves the group or a new member joins the group. Whenever a new member joins the group or an old member leaves the group communication session, the member's key is changed according to the rekeying policy. This rekeying policy is essential since it provides security properties to the applications. For example, the joining members are prohibited to access prior messages – this is known as backward secrecy. Likewise, the leaving members are not allowed to access future messages – this is known as forward secrecy (Kiah & Martin, 2007). Therefore, maintaining an efficient key management system is a challenging task due to the frequent movement of the group members between different Sub Groups (SGs). This triggers the group key adaptation through the rekeying process. The new group key is generated by the server during the rekeying process in order to invalidate the old group key (Mittra, 1997).

Current GKM schemes require updating the key every time if the member moves across multiple groups, which generates significant rekeying overhead. For this reason, these schemes are not scalable for large groups. In a wireless mobile environment, the mobile nodes have limited computational power, storage space and bandwidth and therefore, it is crucial to minimize the computational, storage and communication overheads simultaneously (Rafaeli & Hutchison, 2002). Hence, there is a need to develop a lightweight access control mechanism to address service latency while reducing the rekeying latency concurrently. Figure 1.1 shows a simple example of a group based application, which comprises a key server, group controller and a group of mobile users. The key server is responsible to distribute the keying material necessary to establish group communication. The task of the SG controller is to control the group of

users in order to simplify group communication and facilitate the management of a large group of users.

Secure group based applications can be classified into three schemes (*i.e.* centralized, decentralized and contributory), depending on how the key management tasks are carried out. Centralized schemes are directly dependent on a single main entity to distribute the cryptographic keys. Decentralized schemes are more complex since they involve multiple entities that act as local SG servers which manage a group of users. Contributory schemes involve the participation of all entities in the group creation and key distribution tasks. These schemes are described in detail in Chapter 2.

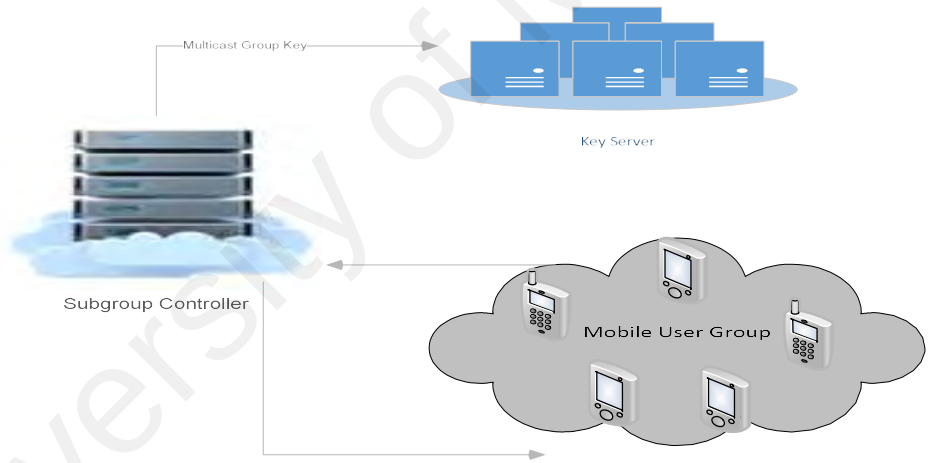


**Figure 1.1:** Group Based Application Scenario

## 1.1 Problem statement

One of the main issues in group based applications is group dynamism. For large groups, the membership changes frequently and each change requires a new key distribution. Therefore, rekeying operation needs to be performed every time a member joins a secure communication session in order to prevent the member from accessing

old messages (backward secrecy) or when a member leaves a secure session to prevent the member from accessing future messages (forward secrecy) (X. Gu, Zhao, & Yang, 2012). In general, mobile nodes are constrained in terms of resources compared to static nodes. This is due to the fact that wireless mobile nodes are smaller, lighter and consumes less power compared to static nodes. However, wireless mobile communication is vulnerable to various kinds of attacks since air is the medium used to broadcast messages over wireless networks. For this reason, ensuring information security is more challenging for wireless mobile communications due to the open nature of the networks and surrounding environment unlike wired networks (Gharout et al., 2010).

In most existing key management schemes (Bruhadeshwar & Kulkarni, 2011; Je, Lee, Park, & Seo, 2010; Zheng, Huang, & Matthews, 2007), the group members obtain a new group key to encrypt and decrypt data for every session update. However, these schemes consume more key computational time and storage and they require additional broadcast messages when there are frequent members joining and/or leaving the group. Tree-based protocols (Liao & Manulis, 2007) reduce the 1-affect-$n$ problem by reducing the overhead of membership change to $O(Log2n)$ for a group of size $n$. These schemes are essentially threshold-based GKMPs, which allows members to compute the group key based on their own participation, which establishes trust between the group users. However, these schemes result in significant computational and communication overheads. In general, hierarchical GKMPs address the scalability issue since the keys are organized in a hierarchical order. In hierarchical schemes (Mittra, 1997; Setia, Koussih, Jajodia, & Harder, 2000), the rekeying operation is introduced only in the SG rather than the whole group. These schemes also introduce additional overhead at the inter-domain level as well as additional information storage in the server, which results

in significant storage costs. Furthermore, the framework of (Kiah & Martin, 2007) suffered from large storage overhead especially for resource-constrained mobile devices owing to the large number of used keys. The common group key is used for all areas which may lead to the 1-affects-*n* phenomenon. The moving member can suffer from join latencies due to rekeying of the area key and common group key. In addition, a member who visits multiple areas may cause the area key and group key to be updated in all areas.

The impact of the rekeying process on group members is commonly known as the 1-affects-*n* phenomenon. This phenomenon refers to the number of group members affected by the rekeying process. The 1-affects-*n* phenomenon is the most challenging problem in designing a group key management framework (GKMF) due to the fact that this phenomenon significantly reduces the performance of the system with an increase in group size. Several efficient GKMPs were proposed in recent years to address these issues. However, these issues increase in complexity if the host mobility scenario is considered in the wireless mobile environment. In a dynamic mobile environment, the GKMPs do not only deal with dynamic group members but also the locations of these members. Whenever a member moves from one SG to another, the key is updated since the member is not known in the new SG area. The number of mobile hosts may be significantly large and more importantly, the mobile hosts may move very frequently. When the mobile user moves from one location to another, the user is perceived as switching from one SG to another, and the keys are updated accordingly while maintaining the group communication session.

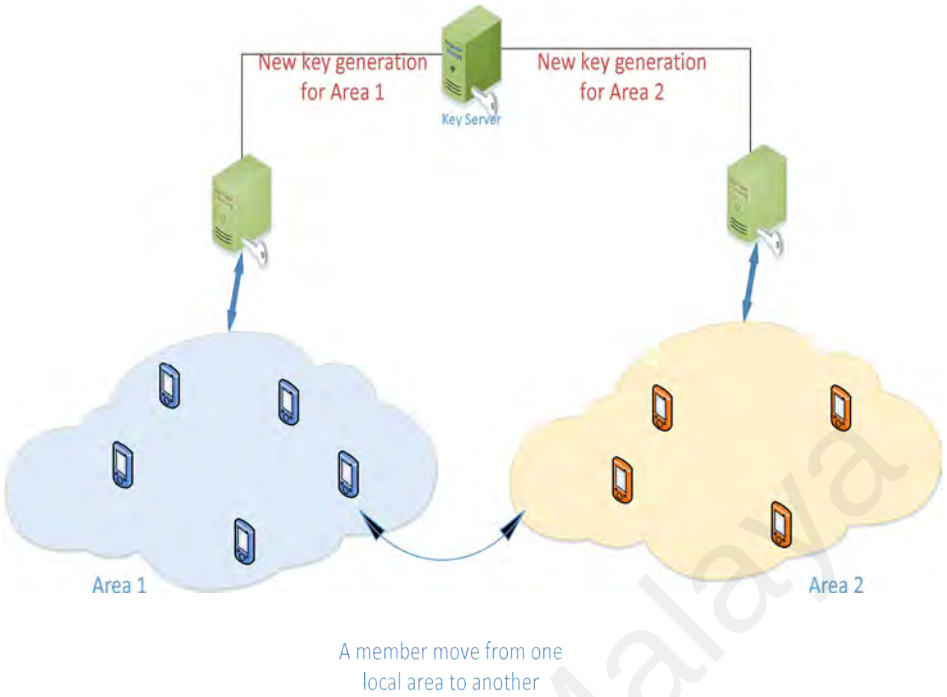


**Figure 1.2:** New Key Generation for both areas upon membership change

In a simple multicast group, the key management overhead may be negligible. However, this is not case for dynamic group based applications since the key management overhead becomes severe due to the considerable number of Traffic Encryption Keys (TEKs) and Key Encryption Keys (KEKs) which are updated constantly across multiple groups. An example of this scenario is illustrated in Figure 1.2, whereby each SG is controlled by a local area or SG, which is typically the case for most GKM schemes. Therefore, if a member moves between several SG, the keying material needs to be updated for all of the affected sub-groups, resulting in significant rekeying overhead. This leads to higher complexity in the key distribution and higher communication cost. It shall be noted that the terms “area”, “sub-group” and “clusters” are used interchangeably in this thesis since these terms have the same meaning.

## 1.2 Research Objectives

The aim of the proposed solution is to analyze, design and implement a lightweight group key management framework by considering host mobility of dynamic group members for group based applications.

Main research objectives of this research work are as follows:

- To carry out a comparative study in existing group key management schemes. This outcome of this study is used to investigate the challenges and issues in current GKM schemes for group based applications.
- To identify the security solutions required for group based applications. This task is used to analyze a suitable key management scheme in wireless mobile environment.
- To develop a proposed solution of key management framework for group based applications. The implementation of the proposed solution is conducted in a network environment by means of simulation.
- To evaluate the performance of proposed solution by considering different security requirements. These security requirements demonstrate the performance of proposed solution by considering storage, communication, and computation of rekeying method.

## 1.3 Research Contribution

This research redirects the existing efforts in the efficient group key management and, a more specifically rekeying method in the wireless mobile environment. The study presents the design, implementation, and validation of new key management solution. The challenges in each SGC framework are analyzed and identified the suitable

methods for the group based applications. The security analysis also considers to show that the proposed framework is secure against many attacks. By addressing the issues of security and performance, our proposed framework aims to accelerate the deployment of group based applications in wireless mobile environment. The propose solution is evaluated through different parameters to estimate the communication, storage, and computation cost of different entities such as mobile members and main server. The experimental simulation results show that the system gains significance improvement in the overhead. Moreover, comparisons of the proposed framework with existing approaches such as (Cao et al., 2006; Gharout et al., 2010; Je et al., 2010; Kiah & Martin, 2007) are conducted by considering above mentioned parameters.

## 1.4 Thesis structure

This study provides the group key management solution for group based applications. The thesis is structured as follows:

- Chapter 2 presented the literature review of the current GKM schemes. Moreover, this Chapter also highlights the challenges in the development of application models and presents a detailed review of the latest group communication frameworks along with their advantages and disadvantages.
- In Chapter 3, the methodology of the proposed framework is presented along with the detail scenario of implementation in NS-3
- The design and implementation of proposed framework are presented in Chapter 4.
- In Chapter 5, evaluation and performance of the proposed key management framework are discussed with security analysis and formal verification.

- Finally, Chapter 6 discusses the conclusion, significance of contribution and future research directions.

# CHAPTER 2: LITERATURE REVIEW

This Chapter presents a taxonomy and survey of the Secure Group Communication (SGC) in wireless mobile environment. The Chapter highlights the characteristics and architecture of SGC in the wireless mobile environment and presents the advantages and disadvantages of different schemes. Moreover, the challenges in each SGC scheme are analyzed to determine that what is suitable for the group based applications. This Chapter compares several approaches and presents the outstanding issues.

In the following Sections, we define some of the key points for group communication architecture. After, we thoroughly explain the group key management schemes and present the taxonomy of these schemes in Section 2.3. Section 2.4 presents the encryption and decryption methods used in GKM schemes. The next Section 2.5 outlines the literature summary of public key cryptography. Section 2.6 describes the group key management methods in wireless mobile environment in addition with the approaches for group communication considering host mobility protocols. Section 2.7 defines the outstanding issues and presents the motivation of this study. The requirements for new GKM framework are presented in Section 2.8 approaches. Finally Section 2.8 outlines the summary of this Chapter.

## 2.1 Definitions and constructions

Group based applications provide proficient delivery of messages from a source to multiple receivers. Before going into in-depth in group key management schemes, we first define some common terms which are used in secure group based applications.

### 2.1.1 Multicast communication

The multicast transmission sends the piece of information to a group of hosts or networks. The data is sent from a single or multiple senders to one or more set of receivers. In Figure 2.1, the scenario of group based applications is shown. This Figure explains how the keys are generated and distributed by the main server to its group members.

### 2.1.2 Key management and distribution

In order to establish the secure group communication, the Key Distribution Center (KDC) generate and send the group key and the users keys in a secure manner. The systems authenticate the users and create and distribute the necessary keys which the users need to communicate with each other. The key management servers are responsible for distribution and key updating in case of any membership changes (Chen & Tzeng, 2017; Islam et al., 2017).

### 2.1.3 Rekeying strategies

The key management protocols are used to devise and employed the group keys in case of membership changes. These protocols provide the authentication services and update the group key in case of any member joins, leaves or member movement within a network group as shown in Figure 2.1. The Figure also shows the process of changing the keys upon each joins or leaves is referred to as rekeying.

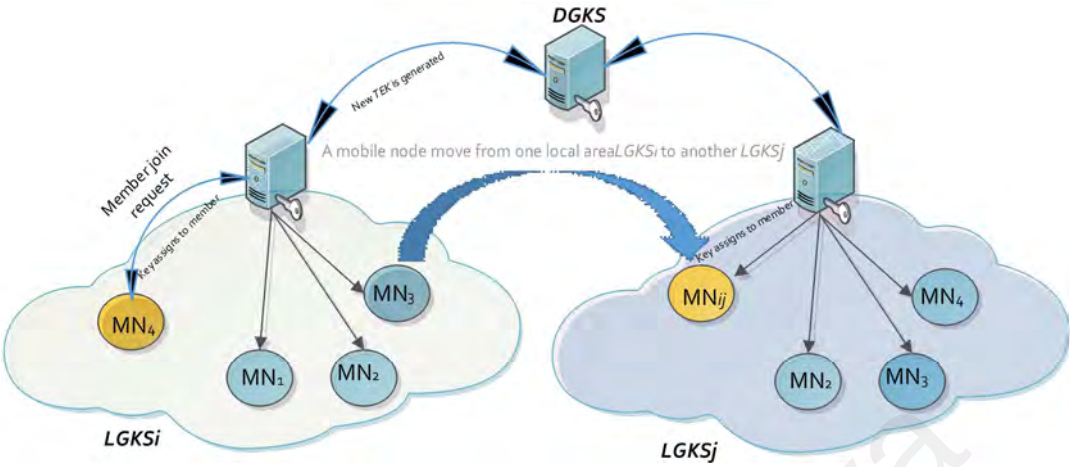


**Figure 2.1:** Rekeying operation in group based application

## 2.2 Security Requirements

The following security requirements should be in place when defining the secure group based application.

(a) ***Backward secrecy***

In multicast scenario, when a new member joins the group, it would not be able to access the previous transmitted information i.e. the joining member would not be able to compute older keys. Every time a new member joins the group, the TEK must be changed for all group members to guarantee that the new joining member cannot decrypt previously transmitted data traffic (Kiah & Martin, 2007).

(b) ***Forward secrecy***

Leaving member from a group cannot be able to access new information in the group. When any group member leaves the multicast session, the TEK must be changed for the remaining group members to guarantee that the leaving member cannot decrypt subsequent data traffic (Kiah & Martin, 2007).

(c) ***Confidentiality***

The message should remain confidential in case of message exchange between different entities in secure group session.

(d) ***Integrity***

The key pairs must be unique and change in every session update to ensure the integrity of the messages.

### 2.2.1 Performance Analysis

The following performance analysis should be considered to demonstrate the effectiveness of the proposed solution.

#### 2.2.1.1 Computational complexity

Computational cost should not include the high number of computation of keys upon every membership change. The computational cost should be acceptable at the server and user side.

#### 2.2.1.2 Communication complexity

Communication complexity indicates the number of messages exchanged during a key generation process.

#### 2.2.1.3 Storage Complexity

The amount of data stored in the key server and in users in order to computes the keys. There should be minimum number of stored keys and data to avoid the high storage cost for key management procedure.

#### 2.2.1.4 Host Mobility Scenario

The "inter-area mobility" states the movement of members between different administrative areas, without leaving or joining the group (Kiah & Martin, 2007).

Several type of group key management schemes are proposed up to now. In general speaking, these schemes can be classified into three types: centralized, decentralized and distributed or contributory group key management as shown in Figure 2.2. Centralized group key management has one central entity which controls the whole group and generates and distributes initial private pieces of information to users in the group. Decentralized scheme has a group and sub group controller which manages the group. Distributed scheme contain no central entity and each group member contributes equally for the creation and the distribution of the group. Secret keys are generated by each member as a function for secure group communication. It is noted that each type of scheme is different from other schemes. In the next section, the detail of three group key management schemes along with their advantages and disadvantages are presented.

## 2.3 Taxonomy of Group Key Management Schemes

This section presents the taxonomy of GKM schemes along with their advantages and shortcomings.

### 2.3.1 Centralized group key management schemes

There are many works in the literature that describes the centralized group key management schemes. In these types of schemes the different users get a new group key which is used for encryption and decryption for every session update. The membership changes with two types i.e. time driven and member-driven.

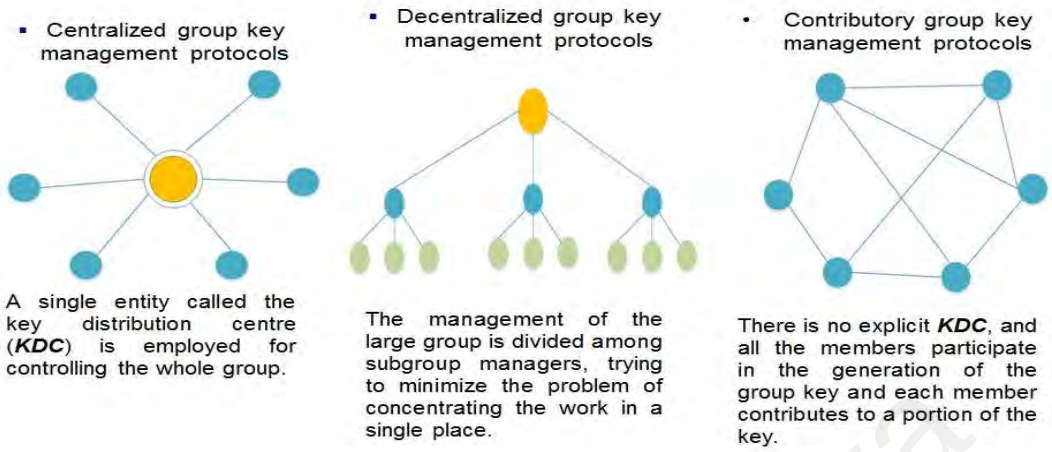


**Figure 2.2:** Group Key Management Schemes

- ***Member-driven schemes***

  In these types of protocols, whenever a member joined or leave the group, the keys should be updated. These schemes can guarantees the forward and backward secrecy (Naor & Das, 2013).

- ***Time-driven schemes***

  In these protocols, the KDC updates the group keys after each regular interval. In time-driven membership the rekeying triggered every interval of time regardless of membership events (Wong et al., 2000).

.

The classifications of member-driven and time-driven approaches are presented in Figure 2.3.

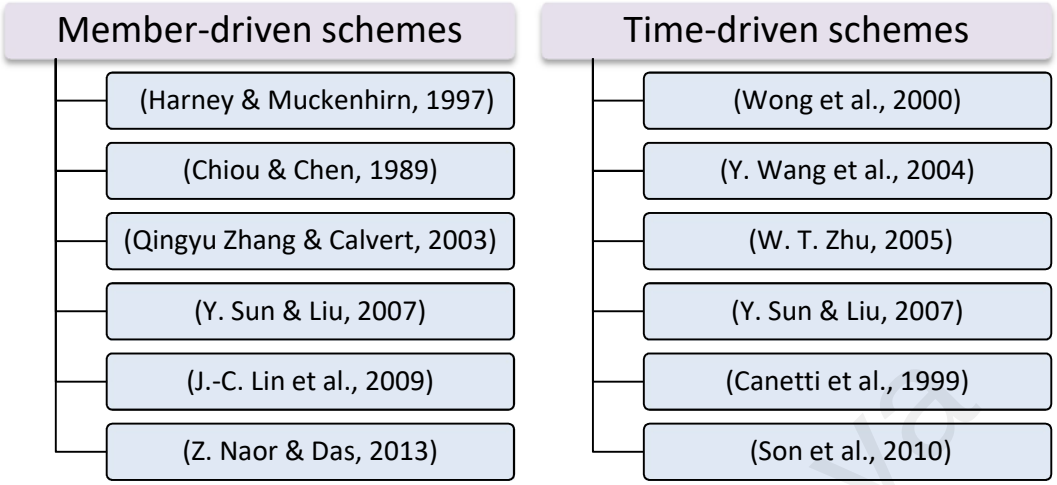


**Figure 2.3:** Classification of member-driven and time driven schemes

In 1997, the work in (Harney & Muckenhirn, 1997) proposed the Group Key Management Protocol (GKMP), in which key server distributes a secret with each member in the group. The KDC generates and distributes the required keys for the members i.e. the group key packet which contains the group traffic encryption key and the group key encryption key. The KDC send a copy of a GK for joining members. Whenever a new user joins the group the KDC generates and distributes the new GK. In the case of rekey, the Group Controller (GC) generates the new GK encrypted with current GK. The disadvantages of this approach are that there is no forward secrecy when a member leaves the group. Moreover, the scheme does not scale to large groups with highly dynamic members.

The work in (Harney & Harder, 1999) proposed the Logical Key Hierarchy (LKH). As can be shown in the Figure 2.4, the KDC maintains the tree of keys and the nodes hold the KEK. Root of the group contains the group key. Tree leaves contains the group

members and each member has a KEK of that particular member. The KEK correspond to the path from its parent leaf to its root.

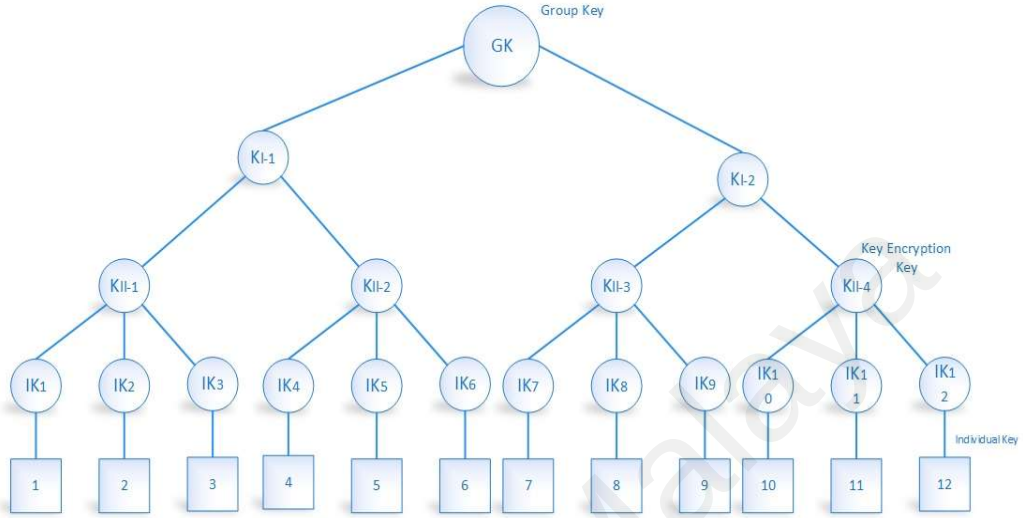


**Figure 2.4:** LKH example consisted of 12 nodes

The method in (Chiou & Chen, 1989) proposed the secure lock in which computational approach is used instead of using the tree arrangement methods. The method based on CRT, and this method required computational on each step of rekey operation. As a result the when the number of users grows the computational time also becomes large.

The work of (Scheikl, Lane, Boyer, & Eltoweissy, 2002) further extended the secure lock method by using the divide-and-conquer approach. This method combines the secure lock and hierarchical tree approach to arrange members in a hierarchical structure. The secure lock method is used to refresh the keys on each level. This arrangement reduces the computation of keys which is suffered in secure lock.

Some of the algorithms are different in joining operation (Waldvogel, Caronni, Sun, Weiler, & Plattner, 1999). These approaches established one way function for members whose keys are affected and need to be updated by the membership changes. Members

can themselves calculate the new key from the old keys. In this way, the new key is calculated locally and there is no need to send the key to the entire group. However, the communication cost has becomes high in this approach for large groups.

The study of has presented by (Wong et al., 2000) to reduce the communication complexity of OFT. They presented a solution for the problem of large groups. The paper constructed the basic key management graph scheme by combining the methods of star and tree base techniques. However, the scalability is still not achieved in this kind of scheme and the computation complexity is also increased.

Another extension of the LKH protocol is LKH++ presented by (Di Pietro et al., 2002). This protocol exploits the properties of both OFT and the set of information that users already share and can be used to generates the new keys in LKH model.

The work in (Qingyu Zhang & Calvert, 2003) proposed the new rekey policy for null rekeying cost. This is called the exposure-oriented rekeying. This proposal based on LKH to achieve the better rekeying cost and security. The authors combine the fact that many auxiliary keys are shared among members in LKH fashion with a hash function. As a result, users are allowed to calculate the new keys by themselves by taking a very little information from the main server.

The work of (Trappe et al., 2003) is based on OFT which is called the Parametric One Way Function based on binary key tree management. KEK is assigned to each node in the system and the members assign the individual key of the nodes with session key. Key updation and distribution is based on the top down or bottom up method. This method uses the embedded keys which is needed to providing the media depending key distribution and therefore increases the computational complexity.

OFT is the extension of the hierarchical binary tree approaches (Sherman & McGrew, 2003). This scheme reduces the size of the rekeys messages. In this scheme, every internal key is build depending on its two descendent keys by using the idea of bottom-top-fashion. This method is more scalable as compared to previous approaches as it considers the computational, communication and storage requirements. In this scheme the internal key is built on its two descendent keys.

In (Penrig et al., 2001) the authors presented the Efficient Large-group Key (ELK) algorithm. This method constructs the logical tree and is very similar to OFT in the way that the parent node keys are generated by its children keys. This method uses the pseudo-random functions (PRFs) to form and maintain the keys in the hierarchical tree fashion. This protocol is the improvement of HTA and is similar to OFT in the arrangement of intermediate keys which are generated from its children. The pseudo random function is adopted here rather than one way function. Therefore, no broadcast messages are needed for joining operations. Moreover, the message loss tolerance policy is used as a hint in broadcast messages so the lost information can be recovered.

Qiong et al. (Qiong Zhang & Wang, 2004) proposed the Enhanced-Hierarchical Access Control (E-HAC). The LKH is formed similar to HAC. However, this method uses the idea of resource group. In this scheme, the data streams encrypted with single TEK so fewer TEK are needed as compared to HAC. The rekeying performance is dependent on the resource group. Consequently, it is very hard to make the resource group by showing the relationship of users. Therefore, it decreases the rekeying performance of the group due to the complicated relation among the group members.

The work in (Y.Wang, Li, Tie, & Zhu, 2004) combined both LKH and one way function. For group rekeying this model used one way function with and old keys and

key management through logical key tree algorithm. The model is more efficient and secure against arbitrary collusion attacks.

The researchers in (Hao, Vinodchandran, Ramamurthy, & Zou, 2005) applied the new scheme which used the leaving tree and based on AVL tree in order to estimate the key tree problem when user departure time is predictable. This scheme is based on individual rekeying patterns and reduces the communication cost.

In order to resolve the problems associated with group key management protocols the work in (Y. Sun & K. R. Liu, 2007) proposed the hierarchical access control technique. The hierarchical access control is the extension of current GKM schemes. In this type of methods, the users have different access right for different data streams. There are users in each SG which are able to access the same subset of data streams. Similarly, there is DG subtree for each DG and their leaf nodes are SG-subtree.

An approach proposed by (J.-C. Lin, Huang, Lai, & Lee, 2009) allows the users to predict the new keys upon membership changes. The least possible amount of information is used from the server. This arrangement reduces the amount of communication and computation overhead. This protocol can handle synchronous and asynchronous protocols and further improved the group key in batch rekey fashion. This protocol combines the encryption and one-way function.

The authors in (Zhou & Ou, 2009) proposed the CRT method which is based on static key structure. The linked list data structure is used to constructs the root ID algorithm. This scenario minimizes the broadcast messages to group of users. Hence the user side key computation is also reduced. However, the workload on the key server is increases to find the common group key by using CRT.

Another effort is made by (Kwak, Lee, Kim, & Jung, 2006) to propose the scheme based on LKH. Whenever the rekeying occurs due to membership change, it changes the structure of the original tree, which is analyzed by this approach.

The work in (Naranjo, Antequera, Casado, & López-Ramos, 2012) presented a new method based on Euclidean algorithm. They present three application of their approach to address security. However, these approaches are the computation complexity in rekeying operation and increase in memory requirements. In addition, complexity of the rekeying operations decreases the performance.

Authors in (G. Xu, Chen, & Du, 2012) proposed the group key management mechanism for delay tolerant network based on Chinese reminder theorem. Hash function is used to calculate the new key from the old group key. Therefore, there is no need that server do not need to update the keys for new users. However, broadcast for user's leaves. In this way they achieve the forward secrecy by introducing the time based group key management.

Authors in (Veltri, Cirani, Busanelli, & Ferrari, 2013) presented the interval-based centralized group key management protocol. The basic idea of this protocol is to predict the user eviction from the group. This task is predicted by the KDC for the duration of the group member's session. The member should leave the group when the period expires without rekeying triggering. However, there are many practical problems such as it is not always possible to predict the exact time of the member leaves event. As a result the approach not well suit for dynamic members with large number of random leaving events such as the in IoT environment. Furthermore, constrained members remain for a longer period of time, therefore, the risk to suffer from storage issues increases.

The method in (Vijayakumar, Bose, & Kannan, 2014) based on Chinese reminder theorem. This is a centralized group key management. They proposed that while distributing the keys the key server only perform one addition operation for updating the group key whenever a new member joins in a group. Similarly, only one subtraction is performed when an existing member leaves the group.

(Varalakshmi & Uthariaraj, 2014) proposed the new method by using the huddle hierarchical method. This approach adopts gray code to provide secure and reliable communication channel. This method reduces the storage overhead both at member side and key server during the group key updating process. Table 2.1 comprehensively presents the literature summary of centralized key management protocols with advantages and disadvantages of each approach.

**Table 2.1:** Literature summary of centralized key management protocols

| Approaches | Advantages | Disadvantages |
|---|---|---|
| (Chiou & Chen, 1989) | The computational approach is used to the problem instead of using tree based arrangement. | Too much computations at the key server on each rekey operation. |
| (C.-H. Lin, 1997) | The ancestor can easily deduce the keys of his descendent. | Once the old key of a group is visible the new key will automatically be exposed. |
| (Harney & Muckenhirn, 1997) | Easy and simple approach. | Infeasible solution for large scale groups. |
| (Canetti et al., 1999) | The members assign on a leaf nodes on the tree. | On member revoked, all the nodes keys should be updated |

| Approaches | Advantages | Disadvantages |
|---|---|---|
| | | and recomputed. |
| (Wong et al., 2000) | Simple multicast key management solution. | Computation complexity is high. |
| (Varalakshmi & Uthariaraj, 2014) | The study targets the multicast batch rekeying operations. | Due to batch joining and leaving, the rekeying cost is high. |
| (Li, Poovendran, & Berenstein, 2002) | Storage efficient scheme. Storage and communication complexity is minimized | Key server and group member computation complexity is high. |
| (Sherman & McGrew, 2003) | This scheme reduces the size of the rekeys messages. | This scheme suffers from the time complexity. |
| (Y. Wang et al., 2004) | The model is more efficient and secure against arbitrary collusion attacks. | Increased in complexity. |
| (Lu, 2005) | Storage efficient key management technique. | Infeasible in the situation where batch leaves are greater than batch join. |
| (Zhu, 2005) | Reduction of the rekeying messages. | Only provide the analysis of communication cost of only A-ary key tree structures. |
| (Zheng, Huang, et al., 2007) | Proposed the hierarchical structure with centralized GKM. | Infeasible for peer-to peer systems. |
| (M. Younis, | Collusion attack probability | Incremental of intermediate |

| Approaches | Advantages | Disadvantages |
|---|---|---|
| Ghumman, & Eltoweissy, 2005) | reduced. | keys. |
| (Y. Sun & K. R. Liu, 2007) | The rekeying cost is reduced. | This method not guarantees the stateless property of GKM protocols. |
| (Zhou & Ou, 2009) | Minimizes broadcast messages required to distribute the group key. | The method increases the workload of key server to find the common group key by using CRT. |
| (Son, Lee, & Seo, 2010) | Reduce the communication overhead. | However, this approach does not provide the authentication capability. |
| (Vijayakumar et al., 2014) | Server only performs one addition/ subtraction operation for updating the group key. | The computational complexity is increases and the scheme suffers from 1-affect-n phenomenon. |
| (Zhao, Kent, & Aggarwal, 2013) | Approach provides the interdependency to each other. | Based on the computationally expensive public key cryptography. |
| (Tsitsipis et al., 2014) | The logical key hierarchy class key management to multi-hop networks. | The computation cost increases for dynamic members of groups. |

### 2.3.2 Decentralized key management schemes

In decentralized group key management architecture, the group is divided into multiple sub-groups. The KDC generates and distributes the group key among all the members. On the other hand, each sub-group has own sub-group key. In this type of communication, the group members involved in a common work activity and they securely communicate and share information among them. In the literature decentralized framework also classified into time-driven and member-driven scenarios. The classifications of these two approaches are depicted in Figure 2.5.

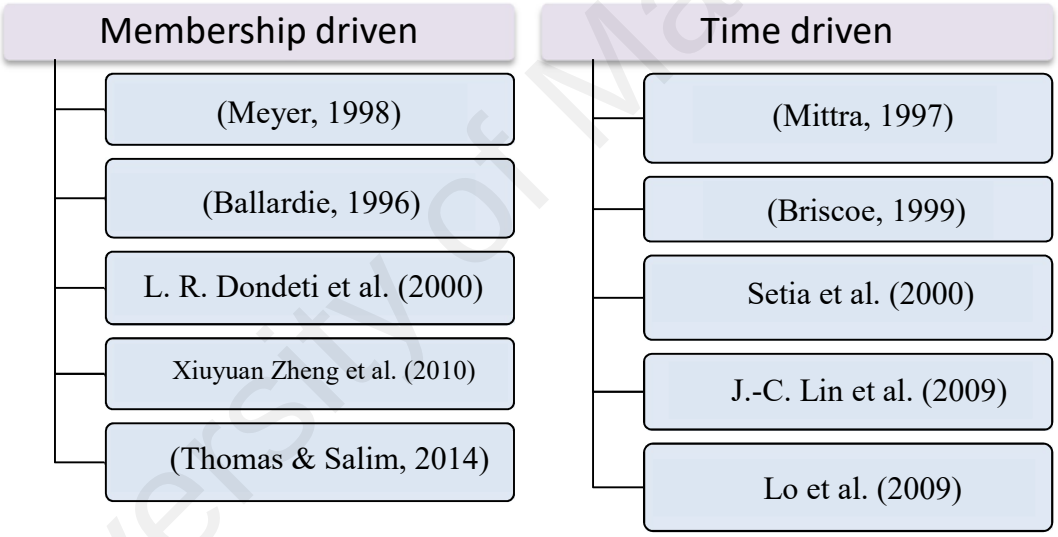


**Figure 2.5:** Classification of member-driven and time-driven schemes based on decentralized architecture

The works in (Mittra, 1997) try to achieve the scalability by splitting the group into hierarchy of sub group's controller that divides the large groups into small group is proposed by mittra by lolus. In this scheme there is a Group Security Agent, which is the controller of each sub group.

Figure 2.6 presented the design of Iolus model in which GSA is grouped into top-level groups. There is no a general group key and membership in each sub group

manages locally. Its means that the changes are treated locally and the changes done in one sub group are not affected by another sub groups. Lolus method is scalable; however there is a drawback that it's affect the data path. The GSA has become the bottleneck whenever its take into account the management of subgroup translation whenever a data goes from one subgroup. The overall efficiency is improved but their support for dynamic group is limited and costly for example it requires intermediary subgroup controller to rely all messages and perform key translation.

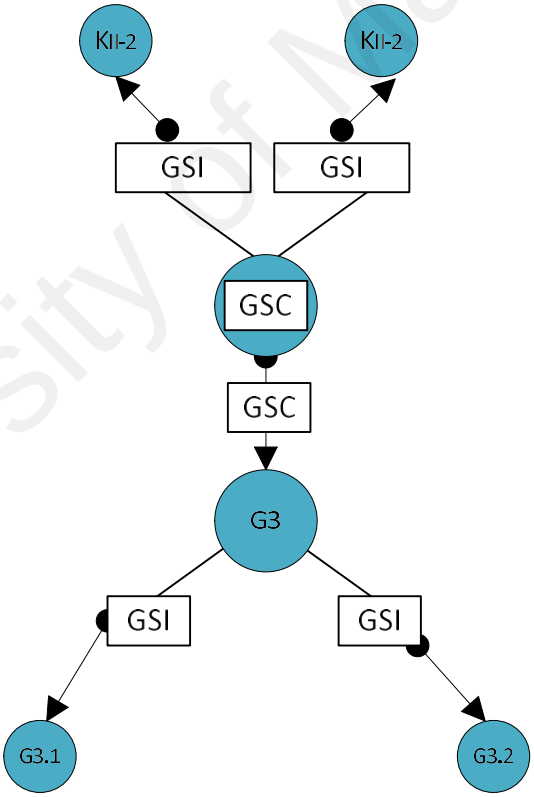


**Figure 2.6:** Iolus framework

The work in (Meyer, 1998) proposed the Internet Group Key Management Protocol (IGKMP). The idea was to divide the whole group into different administrative areas. In this scheme, AKD managed its own area and is responsible of the generation of the group key. As a result it is no need to translate the data packets when its passes from

one SG to another. Moreover, the DKD do contain the list of the group users, these are managed by the AKDs. The central controller DKD can compromise and as a result all the AKDs which are dependent on this can also be compromised due to this central entity.

In 1999, (Briscoe, 1999) proposed the idea to slice the time i.e. small portion of the time and the different keys are encrypted with each slice. Binary hash tree used the encryption keys as the leaves and these keys are generated from a single seed. The tree is generated by using the blinded function such as MD5 (Rivest 1992).

The authors of (L. R. Dondeti, Mukherjee, & Samal, 2000) solved the issue of trusted third-party. They suggested the hierarchical SG of the member where the members are organized by their SG manager. This method produces three types of KEKs and one Data Encryption Key (DEK). This is the kind of hierarchical subgrouping technique of the members where each SG is managed by SG manager. Whenever the data encryption key is needed to be transmitted, the GC generated the package which consists of the DEK and encrypted KEKs.

The approach in (Setia et al., 2000) derives the periodic rekey concept instead of using the rekey policy on membership changes. This scheme generates the group key after a certain period of time regardless of the member's state whether they join, leave or move within a group. This method generates the independently generation of same GK by AKD and sends it to the members at the end of predetermined period. The authors suggest the Network Time Protocol to synchronize the clock. This method do not uses the central entity for key creation, however it creates the keys from the sub group independently, as a result the system becomes fault tolerant because the new key

is generated from the previous key. If one key is disclosed the following keys are compromised.

The authors of (Weiler, 2001) further improved the DEP scheme. This scheme used the dual encryption key for group communication. The forward secrecy is achieved by using this arrangement. However due to the use of dual encryption, the intermediate nodes need to translate the messages, due to which it affect the data path and has the same limitations as the lolus framework.

Further, the work of (Rafaeli & Hutchison, 2002) divided the large group into small subgroups. The server is called the Hydra Server (HS) which controls each sub group as shown in Figure 2.7. As we can see in this Figure, there is not a central group controller. Whenever the membership changes in the specific sub group i.e. HSi the new key is generated by this HSi and send this key to HSj in that session. In this way whenever one or more HSs are not available it still not affects the remaining HSs.

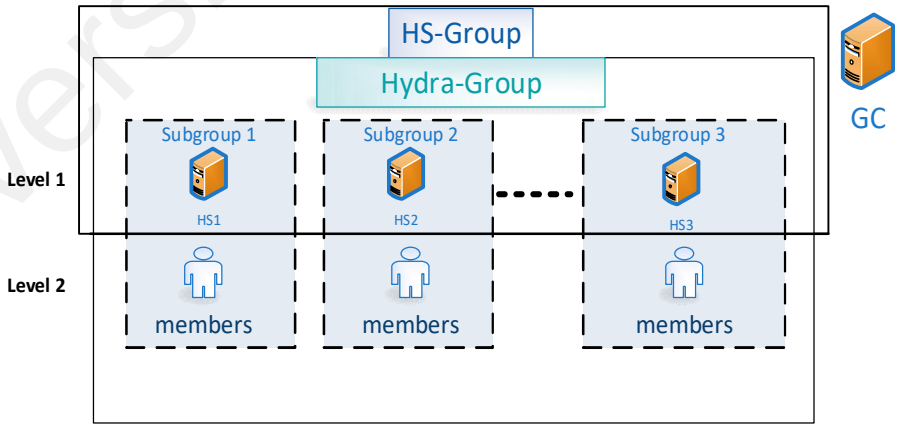


**Figure 2.7:** Hydra system

The study in (Inoue & Kuroda, 2004) proposed the fully decentralized key management scheme called the FDLKH. The scenario of this method is shown in Figure 2.8, in which the key updating mechanism provided for dynamic group without any

central entity. This scheme is based on the LKH. No central entity therefore no single point of failure. It gives the concept of using the representative members who's called the captain using DH key exchange protocol. It uses the binary keys and the intermediate nodes are accompanying with symmetric keys. There are no individual keys for the members. However, they required the numerous sponsors to distribute the keys with the increasing costs of setting up secure channels.

The authors of (Adusumilli, Zou, & Ramamurthy, 2005) proposed the decentralized approach to deal the problem of 1-effect-all phenomenon. The subgroups generate the independent group keys. However these solutions have inefficient due to that they deliver data to their subgroups which performs the decryption and re-encryption for multicasting purposes. Data transmission delays in these operation delays the performance of the system.

The work in (Dutta et al., 2009) introduced the unique assumptions that users cannot revoke the keys before their lifecycle is over. That's means that the implicit users can revoke when user life cycle has over. However, it is very difficult to accurately measure the user life cycle. As a user dynamically added and session key distributed dynamically revoke during system operations.

Vector space secret sharing (Dutta, Mukhopadhyay, & Collier, 2010) is proposed as a key distribution approach to distribute the keys using self-healing method. For access purpose, the Shamir's threshold secret sharing scheme is suggested here. They show that their methods are computationally secure and achieve both forward and backward secrecy. It is believed that these kinds of schemes are only secure for first 't' session.

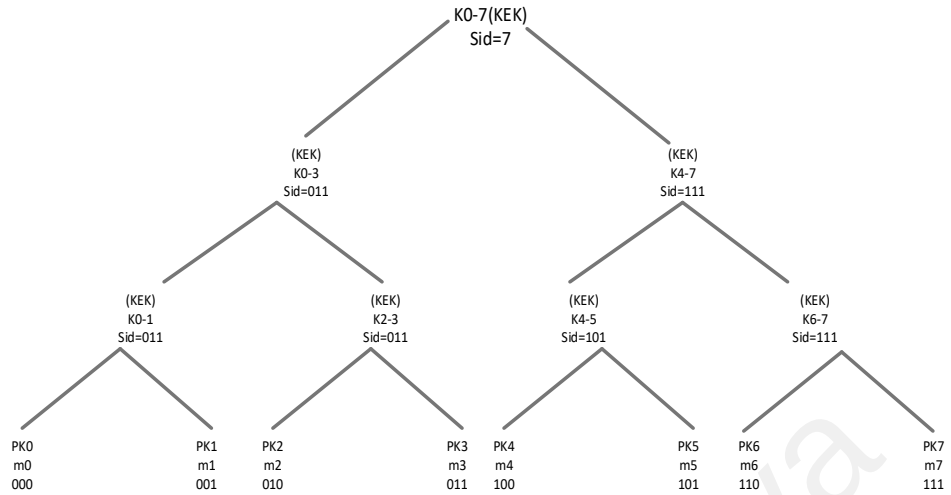


**Figure 2.8:** Sponsor is showing for each node

The authors of (Park et al., 2013) proposed the GKM scheme to consider the multiple multicast in wireless networks. The master-key-encryption based scheme based on the creation of the asymmetric keys such as master keys and multiple slave keys. The master keys are used to create and distributed the cluster keys. Whenever the mobile join or leave the multicast session there is only one key update which is called the slave key and the all the other slave keys remains the same. Only changes the master key.

The authors in (Mehdizadeh at al., 2014) proposed the independent TEK method for each subgroup. When the membership changes, it do not affect the whole groups and only affects the specific SG. However, this method has some limitations such as it affects the data path. Table 2.2 comprehensively presents the literature summary of decentralized key management protocols with advantages and disadvantages of each scheme.

**Table 2.2:** Literature summary of decentralized key management protocols

| Approaches | Advantages | Disadvantages |
|---|---|---|
| (Mittra, 1997) | This method is scalable. | The approach has become the bottleneck whenever data translation occurs from one subgroup to another. |
| (Setia et al., 2000) | The approach derives the periodic rekey. | The system becomes fault tolerant because the new key is generated from the previous key. |
| (Weiler, 2001) | Proposed dual encryption method to provide more security. | It affects the data path, because the intermediate nodes translate messages. |
| (Mehdizadeh et al., 2014) | The method does not need the cluster head. | Introduced too much overhead due to election process of certificate authorities. |
| (Dutta et al., 2009) | Users cannot revoke the keys before their lifecycle is over. | However, it is very difficult to accurately measure the user life cycle. |
| (Inoue & Kuroda, 2004) | Used multiple sponsors and entities for GKM tasks. | Increase in computational overhead. |
| (Adusumilli et al., 2005) | The authors proposed the solution for 1-affection problem. | Data transmission delays in these system delays the performance of the system. |
| (Park et al., 2013) | A scheme considers the | However, the updating of a group |

| Approaches | Advantages | Disadvantages |
|---|---|---|
| | multiple multicasts in wireless networks. | key may cause overhead. |
| (Zhou & Ou, 2009) | This method reduces the user side key computation. | However it increases the workload of key server to find the common group key by using CRT. |

### 2.3.3 Distributed Key Management Schemes

In distributed GKM, there is no main server, and the group members themselves generate the keys. All group members contribute in the creation of the group keys and the user's private keys.

The authors in (Diffie & Hellman, 1976) proposed the well-known key exchange protocol called the Diffie and Hellman protocol. This protocol uses the exponentiation methods in order to form a secret key between two parties. It is influential protocol. Many approaches then proposed to extend the DH protocol which also extends to group key agreement protocols. The group key agreement protocols are based on modular exponentiation these protocols are considered as a variant of DH protocol. Although this protocol is effected by man-in-the –middle-attack.

A CLIQUE (Steiner, Tsudik, & Waidner, 1997) is a protocol to provide the method to key agreement with highly dynamic group member environment. This protocol extends the Diffie-Hellman key agreement protocol.

Later in (Steiner, Tsudik, & Waidner, 1996), the authors extended the DH key agreement protocol. The common key is agreed between two parties. The prime numbers used to calculate the intermediate values through distributive fashion. The first number computes the value and then passes this value to the next number. In this arrangement each member receives the set of intermediate values and generates a new set by calculates them using their own number. However, the proposal effected from storage overhead as the size of the messages increases subsequently as the sequence reaches the last number and there are numerous numbers of intermediate values are needed. The protocol is more applicable for small number of users and cannot used for larger groups.

The study in (Becker & Wille, 1998) proposed the protocol which named as octopus and based on DH key exchange method. They communication complexity is calculated by systematically by using the group key agreement protocols. In order to calculate this they calculate the lower bounds for the number of messages, exchanges, simple and synchronous rounds. The protocols work by dividing the large group into four subgroups. In this scheme, the intermediary DH value is agreed by each SG. After the group members calculates the group key. The responsibility of the leader of each SG is to hold the intermediary DH values which is calculated by each subgroup members. In this way all the group member contributes in calculating of the group key.

Dynamic OFT used the concept of LKH in distribution fashion (L. Dondeti, Mukherjee, & Samal, 1999). There is no any central entity in this method. Every member participates for access control and key generation. Members generate their own keys and make the blinded version of their keys and send it to their sibling.

The authors of (Rodeh, Birman, & Dolev, 1999) presented the distributed approach in which LKH is modeled between the group members and the group controller is eliminated completely. The protocol uses the concept of subtree idea and all agreeing on this mutual key. That implies that two groups agree on a mutual encryption keys.

(Balachandran et al., 2005) proposed the method for key agreement by combining the Chinese Remainder Theorem and Diffie-Hellman (CRDTH) scheme for SGC over MANETs. The key management based on contributory-based GKM. The contributed keys are shared among group members based on Diffie-Hellman. The members are bale to compute the group key based on CRT.

In (Liao & Manulis, 2007), authors proposed the method to combine STR and TGDH protocols to optimize the computation and communication cost efficiency. Tree based protocols reduces the 1-affects-n problem by reducing after a membership change to O(Log2n) in a group size n. This approach provide the threshold based group key management protocols which allows the members to compute the group key based on their own participation resulting the trust between group users. However the member synchronization within a tree is still an issue when failure occurs. And still the computation and communication overhead is large.

The authors (Striki at al., 2006) presented the robust version of TGDH protocol. However, this protocol is impractical for large groups and formed the high number of groups. As a result, there is the waste of resources. This approach uses the concept of "sponsor" to handle the dynamic changes in the key tree. The intermediate key has been generated and distributed for any event occurs due to join, leave and partition of the members.

Anonymous ID-based GKM method is presented in (Wan, Ren, Lou, & Preneel, 2008) for wireless environment. This scheme successfully defends against passive and active adversaries. However it cannot exclude the insider attacks. The insider attacks are handled by (Wu, Tseng, & Yu, 2011) whose proposed the authenticated group key protocol.

The protocol in (Lee, Lin, & Tsai, 2009), is a group key agreement protocol which is based on the bilinear mapping. The security is achieved by using the bilinear computational DH assumptions. However, the protocol is not secure for any adversary and can agree a legal user with other legal user. This protocol also not provides the implicit authentication capabilities.

The cluster based hierarchical algorithms puts (Jabeenbegum, Purusothaman, Karthi, Balachandar, & Arunkumar, 2010), the load of key management between clusters and f dummy nodes without revealing group messages to them to provide the better security.

Vector space secret sharing (J. Gu & Xue, 2010) approach is used to employ group key distribution scheme. They solved the problem of backward secrecy. The difference is that the keyed permutation is done by cryptographically one way hash function.

The key update distribution algorithm is defined in the paper (Kulkarni & Bruhadeshwar, 2010). The system only updates the keys to the users who need them. The algorithms based on the decedent tracking scheme.

The advance hierarchical key management scheme (John & Samuel, 2010) used the stable and power efficient cluster management technique are used for this purpose. The advantage of this method is that it can reduce the overhead on the central server and

need for storing all the keys on the central entity thus reducing their storage overhead on the central entity.

The proposal in (H.-M. Sun et al., 2015) presented the GKA protocol for mobile environment based on short certificateless scheme for authentication purposes. The scheme provides the twofold benefits. First it reduces the cost which is associated with certificate management schemes and secondly the key escrow problem is avoided by the use of certificateless public key cryptography. Table 2.3 comprehensively presents the literature summary of distributed key management protocols along with their advantages and disadvantages of each scheme.

**Table 2.3**: Summary of distributed key management protocols

| Approach | Advantages | Disadvantages |
|---|---|---|
| (Liao & Manulis, 2007) | Reduces the 1-affects-n problem | Takes many rounds of consultations between the nodes. |
| (F. Wang, Yu, & Srinivasan, 2009) | Provide secure solution by considering large integer factorization problem. | The approach suffered from attacks such as private key revealing attacks. |
| (Z. Liu, Ma, Huang, & Moon, 2009) | Low computational and storage overhead. | Not viable for arbitrary large network. |
| (H.-M. Sun et al., 2015) | Provides message confidentiality, digital signature schemes. | Protocol requires slightly more communication bandwidth. |
| (Nam, Lee, Kim, & Won, | Proposed decisional Diffie-Hellman protocol for security | Users do not able to contribute in in group key agreement process. |

| Approach | Advantages | Disadvantages |
|---|---|---|
| 2005) | of mobile users. | |
| (H.-M. Sun et al., 2015) | This protocol reduces the cost associated with and key escrow problem. | This scheme does not provide forward secrecy. |
| (Mortazavi, Pour, & Kato, 2011) | Combines the Diffie-Hellman and symmetric algorithm. | Modulo operation takes long time in key computation due to large key size. |
| (F. Wang et al., 2009) | Security condition is used to select the best available node. | High energy used in key management process. |
| (Sharma & Krishna, 2015) | Less number of keys are needed at join and leave operations. | The computation overhead is increased by the use of both symmetric and asymmetric cryptosystem. |

In the above section we have differentiate the schemes between centralized, decentralized and distributed schemes. All these three schemes suffer from some of the problem such as 1-affect-n, encryption/ decryption and heavy overhead as shown in Figure 2.9.

However, there are some common construct methods which are also used in group key management protocols regardless of these schemes. These common constructs are described are as follows. In the following sections we will see the scheme which based on different methods such as flat table, self-healing schemes, pre-distribution.

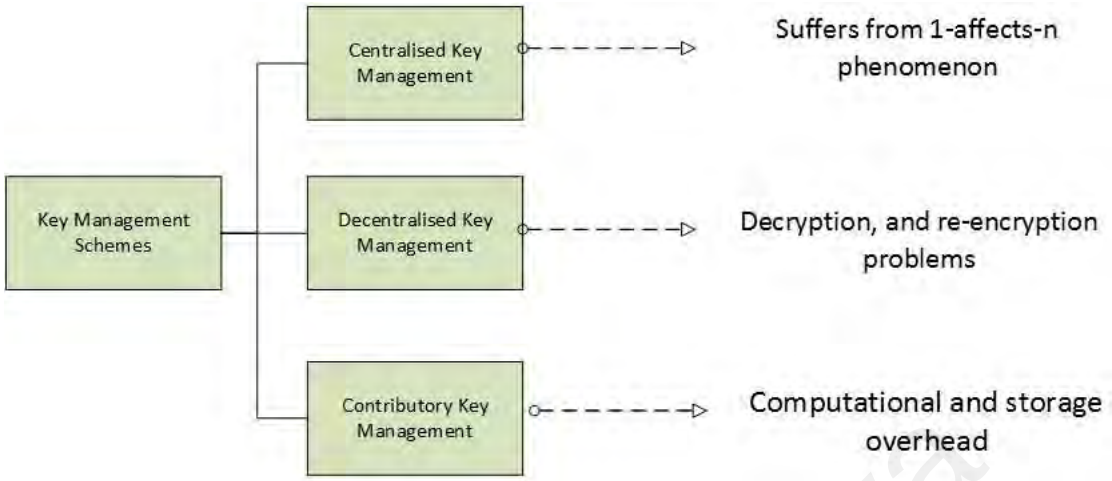


**Figure 2.9:** Issues of Group Key Management Schemes

There are some other group key management schemes exist whether they exists in centralized, distributed or decentralized schemes are described below.

### 2.3.4 Self-healing scheme

Self-healing is the method in which the group members allowed to recover the lost session keys which they could not compute itself because of loss of the packet (Dutta et al., 2010).

### 2.3.5 Group Key Agreement

Group key management protocols allows a set of participant to derive a common secret based on each one's contribution over the open network without relying on any central authority (Kim, Perrig, & Tsudik, 2004).

### 2.3.6 Key Pre-Distribution schemes

This is the type of key agreement scheme, where key information is distributed among all nodes prior to deployment (Camtepe & Yener, 2004; Z. Liu et al., 2009; Sanchez & Baldus, 2005).

### 2.3.7 SGC based on network structure

Besides using probabilistic method, some approaches use the location information for key management (Anjum, 2006; I.-R. Chen et al., 2006; M. F. Younis et al., 2006) instead of using the hierarchical structure. Classifications of these schemes are shown in Figure 2.10.

### 2.3.8 Batch based systems

Batch rekeying (Ng, Howarth, Sun, & Cruickshank, 2007; Pegueroles & Rico-Novella, 2003; S. Xu, Yang, Tan, Liu, & Sesay, 2005) are the methods in which the joining and departing members are associated during a time period before rekeying is performed.

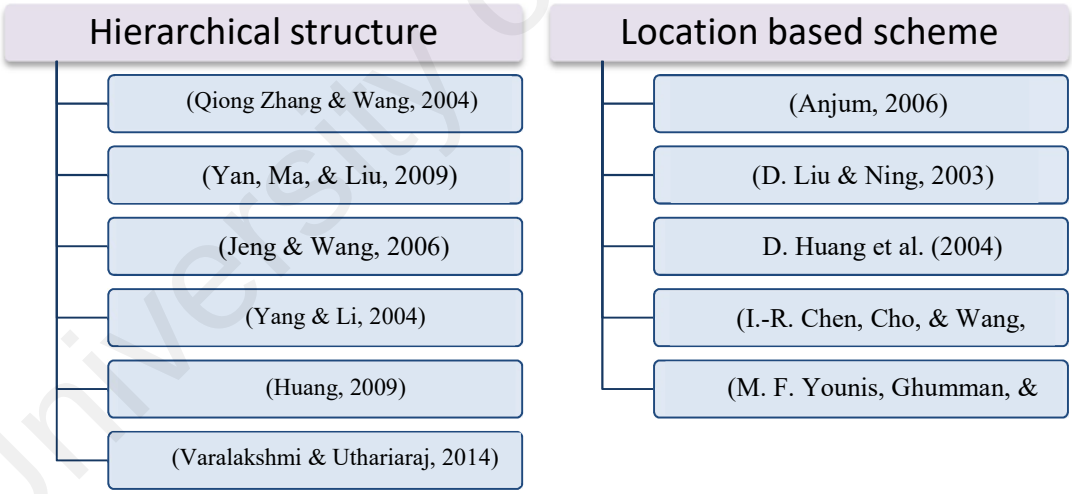


**Figure 2.10:** Hierarchical and location based schemes

### 2.3.9 Group key management based on Flat table

The work in (Waldvogel et al., 1999) proposed the use of flat table with no group controller in distributed schemes. The members don't know all the keys at the same time. The members only know the KEKs. However the distributed key management is

not a feasible idea as the joining member must contact the group members in order to know the all the keys required for group participation.

We have discussed various approaches of group key management is several aspects such as hierarchical, network dependent, network independent, location based etc. We analyzed that each of the protocol like centralized, decentralized or distributed has their own unique features. The centralized approach is easy to implement. The decentralized approach provides the scalable structure by dividing the whole group' participant into sub groups order. And the distributed approach allows every participant to participate in the generation and distribution of the keys.

## 2.4 Group key management schemes in wireless mobile environments

The wide range of mobile devices are becoming popular from last decade, ranging from powerful laptops to handheld Personal Digital Assistants (PDAs) and low power and less computing capabilities mobile devices. The dynamic aspect of the group based application make it difficult due to member's mobility of frequent joins and leaves in addition with member's mobility. Thus it affects the efficient and scalable design of GKM protocols. The challenges are designing the secure and scalable GKM protocols which deal with member's frequent joins and leaves and dynamic member's update due to mobility. In this research work, we present our solution for group key management with a mobility support. Figure 2.11 shows the mobility scenario of mobile users in wireless mobile environment. In this Figure, user 3 which is currently belongs to area 1 is shifted to area 2 while maintains the group session. This scenario is called the inter-area mobility or host mobility.

Wireless technologies and mobile networks have evolved significantly over the last few decades. However, very few studies have addressed the key management in

wireless environment considering host mobility. In the following section, we will studies the key management schemes in the context of wireless mobile environment by considering the host mobility.

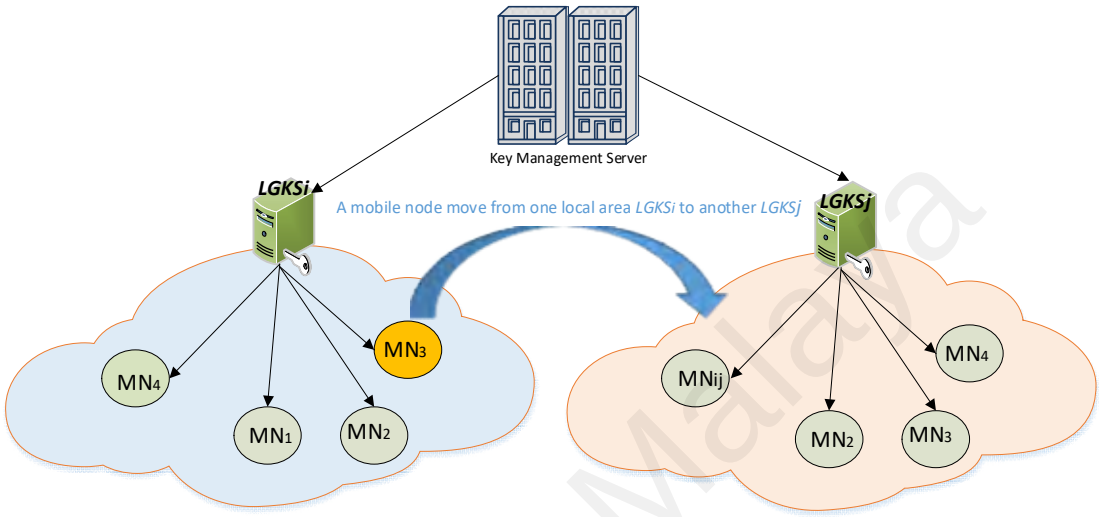


**Figure 2.11:** Inter-area mobility of a node

As we have studied in the previous sections, most of the existing GKM methods were designed for secure multicast communications. Recently, as wireless networks environment become the research hot topic, several researchers have proposed customized GKM for wireless mobile environment by considering host mobility.

In recent years several group key management protocols are proposed to decrease the rekey overhead from both the KDC and on the user. One of the prominent approach is the (Harney & Harder, 1999) logical key hierarchy for scalable group rekeying and to reduce the complexity of rekeying operation to O*(*log *N)*, where *N* is the size of the group. Further it has also been investigated that to do the rekey on periodic basis instead

of every membership change. This method can save the processing and communication complexity of the key server.

ELK protocol (Penrig et al., 2001) is also used to increase the scalability problem of the key server. In addition, it uses the hints in order to recover the lost keys are embedded into the data packets.

The method of combining the star and tree based techniques are presented in (Wong et al., 2000) for the problem of large groups. The paper constructs the basic key management graph scheme by combining the methods of star and tree base techniques. However, the scalability is still not achieved in this kind of scheme and the computation complexity is also increased.

The hierarchical framework is proposed in (DeCleene et al., 2001) to provide the solution for dynamic mobile environment. The authors studied that how the trust and relationships are transferred when members moves across areas in a hierarchy. The central server is called the Domain Key Distributor and is responsible of the key generation, distribution and creation of keys to local controller called the Area Key Distributors. AKDs distribute the keys under their area. The keys remain unchanged whenever the member from area*i* to area*j*, thus movement of member triggers the null rekeying in both of the areas.

Another extension of the LKH protocol is LKH++ presented by (Di Pietro et al., 2002) for mobile wireless networks. This protocol exploits the properties of both OFT and the set of information that users already share and can be used to generates the new keys in LKH model. The method also the case of multiple joins. However, the method not fully treats the mobility issues of the members.

The work in (C. Zhang, DeCleene, Kurose, & Towsley, 2002) provides the hierarchical framework for dynamic environment. The authors studied that how the trust and relationships are transferred when members moves across areas in a hierarchy. The central server is called the Domain Key Distributor and is responsible of the key generation, distribution and creation of keys to local controller called the Area Key Distributors. AKDs distribute the keys under their area. The keys remain unchanged whenever the member from area*i* to area*j*, thus movement of member triggers the null rekeying in both of the areas.

The study in (Chu, Qiao, Nahrstedt, Wang, & Jain, 2002) proposed the idea for message driven protocol. Its means the rekeying occurs for one message. The group member generates a TEK any time its wants to multicast a message. After its share this TEK encrypted with KEK to the group leader. The group leader aims to decrypts it in order to get a TEK and encrypts it again in order to generate KEK to share it with every member on the group.

In order to reduce the effect of 1-affect-n phenomenon the m-lolus (Kamat, Parimi, & Agrawal, 2003) method which is the advanced version of lolus protocol is proposed. This scheme divides the subgroups into micro-subgroups. However, the number of encryption areas increases. The key is not updated whenever a member moves from one micro-subgroup to another. However the scheme violates the backward secrecy.

The method (Hernandez-Serrano, Pegueroles, & Soriano, 2005) used the clustered areas for wireless mobile networks. The method uses the LKH for intra-area rekeying and uses the inter-area rekeying when the mobile member uses from area to area. The method also uses the backbone in order to use the current group key management.

The SHKM scheme (Cao et al., 2006) deployed the subgroups into hierarchical structure with different priorities. The priority of the cluster head is higher as compared to the priority of the local users. These users can be defined as a level of positions. The users of the higher priorities are capable of deriving the keys of their lower priorities users. However the revers operation is not possible.

The work in (Roh & Lee, 2006) designed a two key management schemes for mobile multicast scenario. The method matches the key management tree to the mobile multicast environment in order to localized rekeying messages. However, the scheme is suffer from single point of failure.

The work in (Kellil, Olivereau, & Janneteau, 2004) proposed the mobility issues in mobile multicast scenario. A special key called a visitor encryption key (VEK) is used to represent the moving members. The scheme is based on decentralized method. The Domain Group Controller Key Server (GCKS) is responsible for the generation and distribution of the TEK to the Local Group Controller Key Servers (local GCKS). The Local GCKS is distributed this TEK with specific KEK for each area. Two lists are maintained for members who are called Extra Key Owner List (EKOL) and Visitor Key Owner List (VKOL). The algorithm achieves the null rekeying cost upon membership movement between two areas. This scenario ensured the forward and backward secrecy. The approach implements the common TEK approach due to which it suffers from 1-affects-n problem. This method also does not handle highly dynamic and highly members due to several rekey request. The extra VKOL list also increases the storage overhead to area level which results in processing delays for highly dynamic members.

The hierarchical group access control scheme is proposed in (Y. Sun & K. R. Liu, 2007) to achieve the multi-group key management scheme by constructing logical key

graph by integrating key trees of all members. The scalability achieved by integrating the independent key trees. The access privileges for each member can be achieved by possessed set of keys. The forward and backward secrecy is achieved in this scheme and this scheme requires *O(logN)* communications to update the group key.

The framework of (Kiah & Martin, 2007) addressed the host mobility of the group members in wireless mobile environment for secure group communication. The list is created in order to keep the track of mobile nodes in the network. The domain key manager is responsible for the management and distribution, deletion and updating of the keys and the area key manager is used to perform the key management task in its specific area. Both entities maintain the list which is called the MobList to keep the track of mobility of the members. The MobList also used to maintain the records of the members about the area the member is moving from, ID of the target area. However, this protocol takes the large number of key storage which increases the storage complexity of the resource constrained mobile nodes.

Integrated key graph method for multi-group scenario is presented in (Koo, Kwon, & Ra, 2009). The smaller number of key nodes is used in this scheme. The TEK and user group keys are used. Tree graph is constructed for each resource. Each authorized member stores the set of keys associated with the nodes from the leaf to the root node. The scheme can handle the forward and forward secrecies by changing the access privileges.

The mobility of the mobile nodes are considered in (Gharout et al., 2010) with null rekeying cost. The Domain key manager is responsible for the management of the entire area key manager under it. The AKDs which belongs to the same DKD has common TEK, therefore, there is no rekeying required when mobile node moving from one area

to another. Each AKDi has to maintain the two lists: the list of current member and the list of old members. The protocol is well optimizing the rekey messages by avoiding the TEK every time when the member moves between the clusters whose belongs to the same DKD. Thus, the violation of forward and backward secrecies. The AKD is responsible for the authenticating the nodes thus it's avoid the burden from DKD. Also the member needs to maintain the multiple keys to maintain the sessions.

Authors of (Kwak et al., 2006) presented the scheme based on LKH. Whenever the rekeying occurs due to membership changes, it changes the structure of the original tree, which is analyzed by this approach. They also proposed the optimization method based on LKH.

The authors in (Gharout, Bouabdallah, Challal, & Achemlal, 2012) considered the dynamic groups and treats the nodes mobility with null rekeying and consider the perfect backward and forward secrecy. The protocol considers the independent TEK for each group to avoid the 1-affect-n phenomenon. The protocol is well suitable for both inter-area and intra-area mobility and categories the AKDs belonging to the same clusters or a passive agents. The approach takes the advantage of both common TEK and independent TEK approaches. The area key distributor is responsible for the management of its particular group thus maintain a decentralized approach.

The method in (Park et al., 2013) presented the GKM scheme which considered the multiple multicast approaches in wireless networks. The master-key-encryption based scheme based on the creation of the asymmetric keys such as master keys and multiple slave keys. The master keys are used to create and distributed the cluster keys. Whenever the mobile join or leave the multicast session there is only one key update

which is called the slave key and the all the other slave keys remains the same. Only changes the master key.

The approach in (Trust Tshepo Mapoka, 2013) combines the member authentication procedure with group key management. A simple password authentication protocol is used for member authentication. In the scenario of mobile member moves or leave the group or inter-area mobility the method uses the efficient construct. The method uses the decentralized wireless networks framework with main server distributes multicast content to individual areas. The responsibility of main server is to maintain the list about the member leaves, moves or join the group.

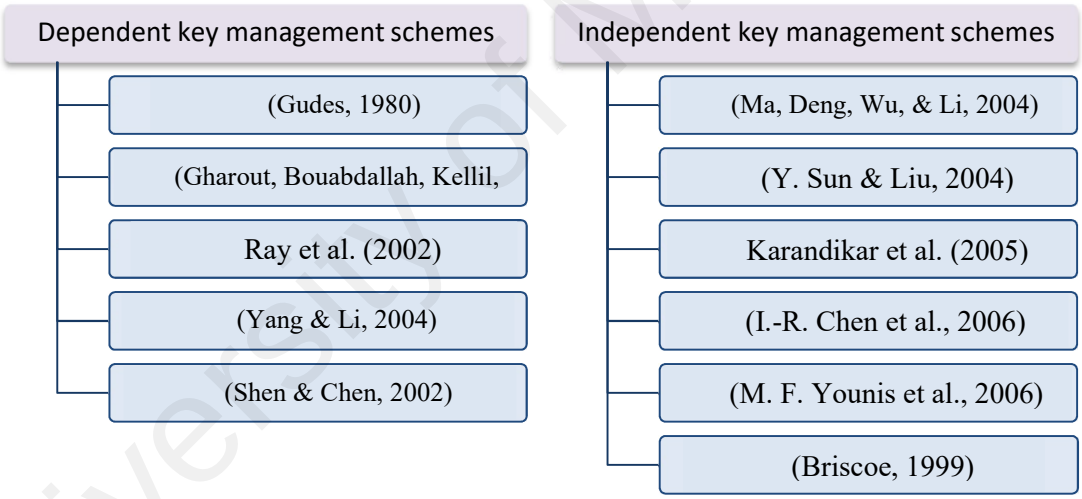


**Figure 2.12**: Dependent and independent key management schemes

The proposal in (Piao, Kim, Tariq, & Hong, 2013) adopts the polynomial-based mechanism to achieve the intra-group key refreshment and to achieve the better scalable for inter-group keys architecture. The intra-group key can be shares between group member and group servers without performing any heavy computations. However the method is not secure because it does not support the intra-group forward and backward secrecies.

Mobility based architecture is proposed by (Madhusudhanan, Chitra, & Rajan, 2015) to address the security issues through the method of rekeying interval. However it introduces the increased in drop of packets rates due to mobility and high number of forward and backward secrecies.

The SMGKM (Trust T Mapoka, Shepherd, & Abd-Alhameed, 2015) scheme provides multi-group based services for mobile users who change their locations dynamically. The mobile users can subscribes to diverse number of services while maintaining the backward confidentiality. The scheme is based on the slot based method for the dynamically members where users wants to subscribes to multiple network services. To alleviate 1-affect-n phenomenon the scheme uses the independent rekeying strategies having the independent TEK for each cluster. The two tier architecture adopts to define the AKD and DKD. The scheme provides the efficient communication and storage overhead at DKD and AKD level. The scheme constructs the list for the members who moves between different areas while maintain their subscriptions to the previous areas. Figure 2.12 classifies the approaches which used the dependent and independent group key management protocols.

**Table 2.4**: Literature summary of key management schemes in wireless mobile environment

| **Approaches** | **1-aff-n** | **KMS** | **KI** | **HM** | **Security** | | **Remarks** |
|---|---|---|---|---|---|---|---|
| | | | | | **FS** | **BS** | |
| (Harney & Muckenhirn, 1997) | ✓ | Cen | × | ✓ | × | ✓ | RO |
| (Harney & Harder, 1999) | ✓ | Cen | ✓ | × | ✓ | ✓ | SO |

| Approaches | 1-aff-n | KMS | KI | HM | Security | | Remarks |
|---|---|---|---|---|---|---|---|
| | | | | | FS | BS | |
| (Penrig et al., 2001) | ✓ | Cen | ✓ | × | ✓ | ✓ | RO |
| (Gharout et al., 2012) | ✓ | Dec | × | ✓ | ✓ | × | FS breach |
| (Amir et al., 2004) | ✓ | Cen | ✓ | × | ✓ | ✓ | CO |
| (Kiah & Martin, 2007) | ✓ | Dec | × | ✓ | × | ✓ | CO |
| (Y. Sun, Trappe, & Liu, 2004) | ✓ | Dec | × | ✓ | ✓ | ✓ | Cmp.O |
| (Kellil et al., 2004) | ✓ | Dec | × | ✓ | × | ✓ | RO |
| (Park, Park, & Seo, 2010) | × | Dec | ✓ | ✓ | ✓ | ✓ | SO |
| (DeCleene et al., 2001) | × | Dec | × | ✓ | ✓ | ✓ | Cmp.O |
| (Hernandez-Serrano et al., 2005) | ✓ | Dec | ✓ | ✓ | ✓ | ✓ | RO |
| (Bouassida, Chrisment, & Festor, 2008) | ✓ | Dec | ✓ | ✓ | × | × | CO |
| (Kamat et al., 2003) | × | Dec | × | ✓ | ✓ | × | RO |

| Approaches | 1-aff-n | KMS | KI | HM | Security | | Remarks |
|---|---|---|---|---|---|---|---|
| | | | | | FS | BS | |
| (Cao et al., 2006) | × | Dec | ✓ | ✓ | ✓ | ✓ | Cmp.O |
| (Park et al., 2013) | × | Dec | ✓ | ✓ | ✓ | ✓ | CO |
| (Trust T Mapoka et al., 2015) | × | Dec | ✓ | ✓ | ✓ | ✓ | Cmp.O |

Symbols and abbreviation used in this table:

Yes=✓, No=×, Decentralized=Dec, Centralized=Cen, Computational Overhead= Cmp.O, Rekey Overhead= RO, Communication Overhead= CO, Storage Overhead= SO, Forward Secrecy= SC, Backward Secrecy= BS, Host Mobility= HM, Key Management Scheme= KMS, Key Independence= KI, 1-Affect-n= 1-aff-n

Table 2.4 summarizes the literature summary of existing GKM protocols in wireless mobile environment, with the detail description of different security features such as backward secrecy, forward secrecy, host mobility support, and 1-affect-n phenomenon.

## 2.5 Encryption/ Decryption methods in GKM

There are many methods used for encryption and decryption in GKM schemes. However, these methods can be divided into two most prominent methods i.e. symmetric and asymmetric encryption.

### 2.5.1 Symmetric and asymmetric algorithms

In symmetric key methods the encryption/ decryption is performed by using the same key. This method is also called the shared key or shared secret encryption. Some of the widely used symmetric methods are AES (Advanced Encryption Standard). Other examples of symmetric encryptions are DES, RC4 etc. Symmetric algorithms are best for their security and high speed (Bellare, Desai, Jokipii, & Rogaway, 1997; Fujisaki & Okamoto, 1999).

In asymmetric encryption different keys are used for encryption and decryption purposes. Public key is used for data encryption which is then decrypted with private key. In this study, we are using RSA and CRT algorithms (Barrett, 1986) for encryption and decryption operations. Table 2.5 comprehensively presents the differences between symmetric and asymmetric encryption methods.

**Table 2.5:** Differences between symmetric and asymmetric methods

| Attributes | Symmetric encryption | Asymmetric encryption |
|---|---|---|
| Examples | AES, DES, Blowfish etc. | RSA, Diffie-Hellman, ElGamal etc. |
| Key type | Shared secret key | Unique public and private key. |
| Strengths | Fast performance and easy to understand. | • Private keys are never exposed.<br>• Exposure to information is limited.<br>• Provides authenticity of the source. |
| Weaknesses | Shared secret key can exposed, | Public key management which is |

| Attributes | Symmetric encryption | Asymmetric encryption |
|---|---|---|
| | and it does not provide authenticity of source. | computation-intensive |
| Key differences | Symmetric cryptography keys are stored in the respective application, and if found can be used to forge software license. | Asymmetric cryptography does not enable hacker to forge the license for others. |
| Functionality | Efficient communication can be guaranteed between two parties. | Allows security is such situations where symmetric encryption cannot provide efficient security and difficult to implement. |
| Computational efficiency | Fast cryptography method due to simple calculation. | Compute slow computations due to heavy calculations. |
| Key size | Uses 128-bit symmetric keys. | Key size should be 1000 bits to achieve sufficient security. |
| Security service provided | Confidentiality | Confidentiality, Authentication and non-repudiation |



## 2.6 Literature of RSA and CRT algorithms

In this Section, the RSA and CRT algorithms are presented in this section. The literatures of both methods are presented with thorough analysis of combining both RSA and CRT algorithms.

### 2.6.1 RSA

The most commonly public key algorithm is Rivest-Shamir-Adleman (RSA) (Goldwasser, Micali, & Rivest, 1988). There are different key used for both encryption and decryption. Public key is used to encrypt the data and the private keys can be used to decrypt the data. For digital signature methods, signature is verified by the public key which is bound to the data. Binding can be accomplished by using the certificate. The RSA cryptography methods are very secure and their security lies in the fact by considering the factoring of numbers. Usually 1024 key size is used for reasonable level of security (Goldwasser et al., 1988; Rivest, Shamir, & Adleman, 1978).

### 2.6.2 Chinese Remainder Theorem

Chinese remainder theorem dated back to 4 AD from an old Chinese puzzle by Sun Tsu Suan-Ching:

"There are certain things whose number is unknown. Repeatedly divided by 3, the remainder is 2; by 5 the remainder is 3; and by 7 the remainder is 2. What will be the number?" In modern number theory, this problem is called as the simultaneous congruence in modern number theory. The Chinese remainder theorem (Shamir, 1979) is used to give a unique solution to simultaneous linear congruence by using coprime moduli. In the basic form, this theorem gives the number, when divided by some divisors, gives remainder. The Chinese remainder theorem often allows us to reduce a question involving modulus $n = p_1^{\alpha_1} p_2^{\alpha_2} \ldots\ldots p_l^{\alpha_l}$ into a question for the prime power moduli $p_i^{\alpha_i}$. Now, we will see the approaches which are based on the RSA and CRT encryption methods (Rivest et al., 1978; Shamir, 1979).

The SHKM (Rivest et al., 1978) uses the RSA method to deploy the subgroups into hierarchical structure. In this method, the cluster head play the important role whose priority is higher than other local users. These users can be defined as a level of positions. The users of the higher priorities are capable of deriving the keys of their lower priorities users. However the revers operation is not possible. By doing this arrangement, rekey overhead for updating the GK can be reduced significantly.

Hierarchical CRT (Zheng, Manton, & Huang, 2007) used the XOR addition and multiplication to broadcast only one rekey message. The member only needs to compute the 1 XOR and 1 modulo erythematic operation to establish the new group key. However, due to this calculation the users need to store the four key which increases the storage cost at the user side.

The work in (Zheng, Huang, et al., 2007) proposed the fast CRT for group key management. The advantages of their protocol are that the number of broadcast messages to distribute the group key to user side minimized. The user side key computation is also minimized. However, the approach is faces the computational complexity of key server which is very high.

The authors in (Shinde & Fadewar, 2008) claims that their method is the four times faster RSA-CRT algorithm for decryption of data. The use of CRT is used for data security.

The approach of (Munivel & Lokesh, 2008) is taking the benefit of CRT and LKH to reduce the rekeying messages. Multicast group is divided into the clusters and each cluster has their own group controller. For inter cluster key management method is done with CRT. The subgroup controller generates the public key and uses the session key of

the members to generates the secure lock which only be decrypted by the cluster members to get the session key.

The work in (Zhou & Ou, 2009) proposed the CRT based static key algorithm for generating and distributing the group key to group members. The construction of the root ID is done by using the linked list of data structure to minimize the computation overhead on user side. However the major drawback of this approach is that it increases the load on key server to find a common group key by using the CRT for *n* number of congruential equations.

The paper in (Zhou & Ou, 2009) proposed the CRT based static key method for generating the GK to members when there is change in the group membership. The liked list data structure is used to construct the ID construction algorithms which can be minimizing the messages to distribute the GK to group members. By using this method the user's side key computation is also reduced. However, the method also increases the load on the main server by calculating the common GK for 'n' number of congruential equations.

A GKM (Park et al., 2013) scheme is proposed by to consider the multiple multicasts in wireless networks. The master-key-encryption based scheme based on the creation of the asymmetric keys such as master keys and multiple slave keys. The approach uses the CRT to establish the master key. Master keys are used to create and distributed the cluster keys. Whenever the mobile join or leave the multicast session there is only one key update which is called the slave key and the all the other slave keys remains the same. Only changes the master key.

The paper (Y. Xu, Zhou, & Wang, 2014) proposed the approach which is based on the CRT to define the encryption rekeying to a short term messages. The key graph

method is used to construct multiway trees. The method attained the backward and forward secrecy. The multiway trees are used to reduce the height of the tree and reduced the number of intermediate nodes. However, the computational cost is increases due to using the two methods i.e. CRT and one-way function.

The work in (Y. Liu, Harn, & Chang, 2014) proposed to establish the session key of group by combining the threshold secret sharing method and CRT. This method distributes the confidentiality of the group members and shares the secret with each group member. Each member of the group shares his secret with key server to recover the group key.

The paper in (Thangavel, Varalakshmi, Murrali, & Nithya, 2015) proposed the method which is based on RSA cryptosystem. By using the RSA methods the time required for cryptanalysis to encrypt a message is higher. However, by using the RSA method the system becomes very safe and cannot breakable easily. Through comparison the author claims that their scheme is efficient as compared to traditional RSA scheme. However, this scheme takes four large prime numbers, as a result the system complexity increases as compared to simple RSA algorithm.

## 2.7 Discussion and motivation

In this Chapter, a comprehensive review of the literature relevant to this research is presented, with emphasis on the classification of GKMPs and algorithms. Most of the GKMPs are designed for wired and wireless networks and therefore, they are not applicable for wireless mobile environments. Wireless mobile environment is intrinsically more complex because it does not only involve the join or leave of the members, but also the movement of the members between areas. This in turn, increases the complexity of the GKM. As discussed in this Chapter, the key management

techniques can be divided into two types: pairwise and LKH tree based approaches. In this study, the comparison of proposed solution is mainly conducted with pairwise approaches such as (Gharout et al., 2010; Kellil et al., 2004; Kiah & Martin, 2007; Waldvogel et al., 1999; C. Zhang et al., 2002; Qingyu Zhang & Calvert, 2003). In addition with, we also perform the comparison with LKH (Canetti et al., 1999) based rekeying approaches in order to demonstrate the effectiveness of the proposed solution.

A couple of approaches have been developed (Daghighi, Kiah, Iqbal, Rehman, & Martin; Gharout et al., 2010; Kellil et al., 2004; Kiah & Martin, 2007; Waldvogel et al., 1999; C. Zhang et al., 2002; Qingyu Zhang & Calvert, 2003) to address group communication in wireless mobile environments. However, there are shortcomings with these approaches, particularly in handling group dynamism as well as the significant rekeying overhead. According to DeCleene et al. (2002), service disruption occurs in both new and old areas during the rekeying process. The Batch Rekey (BR) (C. Zhang et al., 2002) scheme is incapable of differentiating between each join and leave of a member when the member moves across the areas. The Immediate Rekey (IR) (C. Zhang et al., 2002) scheme suffers from significant rekeying overhead, especially if the member moves rapidly across the areas. This problem needs to be addressed by repeated local rekeying. The Delayed Rekey (DR) (C. Zhang et al., 2002) scheme is disadvantageous since backward secrecy does not occur until the next member joins or leaves. In the Periodic Rekey scheme, data transmission is not possible until all of the visited areas are rekeyed and a new key is distributed. This result in delay and the members remain offline, unlike IR. First Entry Delayed Rekeying + Periodic (FEDRP) (C. Zhang et al., 2002) scheme appears to be the most desirable approach because of its low communication overhead and high scalability. Moreover, this scheme supports highly dynamic membership. However, the disadvantage of this scheme is its area key

which can be compromised easily. In addition, the FEDRP, GKMF and Kellil et al.'s schemes suffer from the 1-affects-*n* phenomenon since TEKs are used in the key management task. In these schemes, each join or leave of a member may lead to rekeying problems. Hence, these schemes are incapable of handling high dynamism and mobility of the group members due to the multiple rekeying overhead. In Gharout et al.'s (2010) scheme, the moving member may experience join latency due to delays in data transformation processing when the member inter-moves. In brief, the schemes proposed in previous studies for wireless mobile communications suffer from the same fundamental problem: rekeying overhead.

Moreover, the decryption process of GKMPs in resource limited mobile devices can be expedited by implementing and efficient GKM algorithm such as Chinese Remainder Theorem with Rivest-Shamir-Adleman algorithms compared with modular exponentiation. The RSA-CRT algorithms differ from standard RSA algorithms in terms of key generation and decryption. In RSA-CRT algorithms, the RSA encryption accounts for the heavy load of calculations during the encryption process by placing the heavy load at powerful servers whereas the CRT algorithm accelerates the decryption process at resource-limited mobile devices. Hence, the benefits of RSA-CRT algorithms (*i.e.* maximum security, confidentiality and authentication) can be exploited for critical group based applications where security is of utmost importance.

However, the main challenge is to attain an ideal balance between security, computational complexity and communication and storage costs. These issues are not tolerable for applications involving dynamic users and resource-constrained mobile devices. In addition, the current GKMPs are not scalable, which presents additional challenges since it is crucial to reduce the significant overhead as much as possible resulting from frequent join, leave and switch operations. Common TEK solutions

suffer from the 1-affects-*n* phenomenon whereas the Independent TEK solutions suffer from a significant number of decryption / encryption operations which need to be addressed. Cryptographic algorithms play a vital role in designing lightweight group key management framework. More importantly, current GKMPs are incapable of resolving issues associated with dynamic group members, which are known to degrade the performance of the application in highly complex, unstable wireless mobile environments.

## 2.8 Requirements for new group key management framework

Since the structure of a wireless mobile environment is very close to that for decentralized GKM, a decentralized architecture is adopted in this research to design a lightweight key management framework, taking into account group dynamism. In the proposed framework, the whole group is divided into different sub-groups and each sub-group maintains its own members and keys. The key management framework is developed based on the following requirements:

- The whole group must not be affected from a single member change to prevent the 1-affects-*n* phenomenon. This task is executed when a single member changes, which requires all of the group members to commit to a new group key.
- The RSA-CRT algorithms are chosen for the key management framework, which accelerates the key management tasks by increasing the computational speed. The CRT algorithm improves the performance of the RSA decryption algorithm at resource-constrained mobile devices.
- For dynamic group members, the bandwidth must not be high for rekeying messages. In addition, the bandwidth must be independent of the group size.

- The proposed solution should reduce the number of keys stored by the users and the GC.
- The proposed solution should minimize the, communication, rekeying transmission and storage overheads, and optimize the signaling load at the central network. There should be no single points of failure and the 1-affects-*n* phenomenon.

## 2.9 Chapter Summary

A comparative study of existing group key management schemes are presented in this Chapter. The characteristics of wireless mobile networks, and the architecture and entities that affect the distribution and management of keys are presented. This Chapter also presents a qualitative comparison of existing key management schemes in order to obtain a better understanding on the features, advantages and disadvantages of each protocol, with emphasis on host mobility issues in wireless mobile environments. The limitations of current key management schemes are to identify the requirements for group based applications. These understanding helps to design a suitable key management framework for group based applications

# CHAPTER 3: RESEARCH METHODOLOGY

The Chapter outlines the research methodology adopted in this study. The research methodology helps to execute the research in this study. The stages involved in this research work are presented in Figure 3.1. This Chapter is organized by presenting the outcomes of problem finding from the literature review in current key management framework are Section 3.1. Section 3.2 presents the system design adopted in this study. The implementation detail of proposed framework is presented in Section 3.3 the methods of security analysis are discussed in Section 3.4 the results gathering and comparison methods are describe in Section 3.5. Finally, Section 3.6 presents the summary of this Chapter.

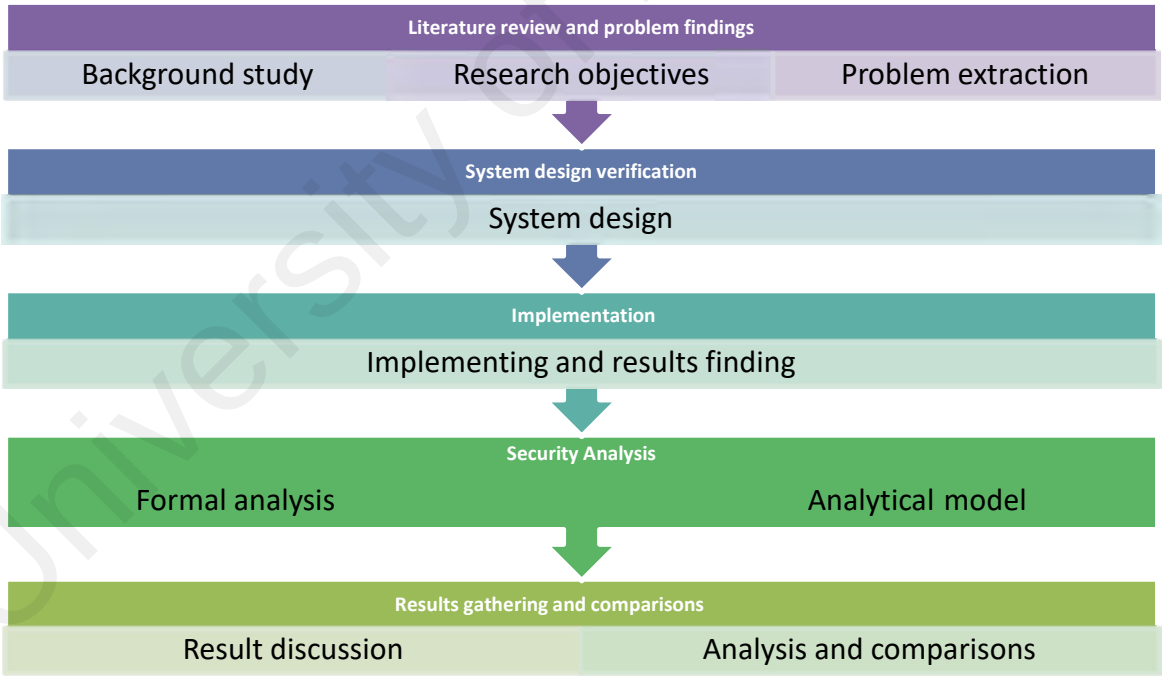


**Figure 3.1:** Research methodology overview

In the following section, the steps involved in the construction of research methodology are explained briefly.

## 3.1 Literature Review and Problem Findings

The work in this study is to design and implement the secure key management framework for group based applications. Therefore, the following related work in used to explore the existing key management protocol. We have thoroughly investigated that are presented in the literature and what is the research gap in the existing knowledge.

In this study, we conduct a systematic survey of the literature from the following electronic databases:

- Springer (www.springerlink.com)
- Elsevier (https://www.elsevier.com/)
- Taylor & Francis Online (www.tandfonline.com)
- ScienceDirect (www.sciencedirect.com)
- ACM Digital Library (www.acm.org/dl)
- The Scientific World Journal (www.hindawi.com)
- IEEE eXplore (www.ieeexplore.ieee.org)

The existing key management protocol normally based on three schemes i.e. centralized, decentralized and distributed schemes. A qualitative comparison has been performed to perform the advantages and disadvantages in each scheme.

## 3.2 System design and verification

We use these resources to extract the problem related to the latest information necessary to design the method for group based applications. We thus review and classify the existing GKM protocols and make taxonomy of the existing group key management schemes. These schemes are centralized, decentralized and distributed key management architecture. We have thoroughly investigated the critical factors of these schemes in order to find the advantages and disadvantages of these schemes. The

qualitative analysis helps to find out the results from these schemes. The qualitative analysis has presented in the form of Table and the behavior of the several group key management systems has been defined. The design and model are used to implement the system in network simulator environment. Moreover, the BAN logic, which is well known formalism analysis of protocol used for authentication is used in this research to provide the security of proposed framework.

### 3.3 Implementation

The implementation has been done of the overall system design by using the open source Network Simulator 3 (NS3) simulator (Carneiro, 2010). The implementation task helps to validate and modelled the design process which we have done in the last phase. In NS-3 simulator, the following function has been used to implement the proposed methods.

**Table 3.1:** Simulation Parameters

| **Parameters** | **Descriptions** |
|---|---|
| Network Simulator | NS3 |
| Mobility Period (sec): | 0.008333 |
| Simulation Time | 10m |
| Internet | LTE |
| Mobility | Mpi |
| Animator | NetAnim |
| Simulation Area | 200m x 200m |
| Number of nodes | 500 |
| Data Type | CBR |
| Transport Protocol | UDP |

| | |
|---|---|
| Packet size | 512 bytes |
| MAC protocol | IEEE802.11 p |
| MAC Rate | 2 Mbps |

### 3.3.1 LTE Model

Long-Term Evolution (LTE) is the standard for high speed wireless data communications technology developed from GSM/ UMTS network technologies. By using these network technologies can increase the speed of using the different radio interface. This model is able to provide the following functionalities of LTE systems:

- Radio Resource Management
- Dynamic Spectrum Access
- Inter-cell Interference Coordination
- QoS-aware Packet Scheduling

The LTE architecture of high-level components can be shown in Figure 3.2:

- The User Equipment (UE)
- The Evolved UMTS Terrestrial Radio Access Network (E-UTRAN)
- The Evolved Packet Core (EPC)

The interfaces between the different parts of the system are shown in Figure 3.2.


**Figure 3.2:** LTE Network Architecture

### 3.3.2 User equipment: UE

The main goal of the LTE technology is to provide the high speed infrastructure to their users which 3G networks cannot. In LTE architecture the eNBs (evolved NodeB) is responsible for managing the UE's communication by managing the radio resource and mobility in the cell. The LTE eNB component depends on the radio resource management algorithm. In LTE architecture UE is like a device which is used by the end users for communication. These devices can be laptop, hand-held telephone or any other smart device. It can be connected to the base station NodeB/eNodeB as specified in the ETSI 125/136-series and 3 GPP 25/36-series of specifications. In GSM system this can be considered as mobile station in GSM systems. There are following important modules in mobile equipment:

- Mobile Termination (MT)
- Universal Integrated Circuit Card (UICC)
- Terminal Equipment (TE)

UE handles the following tasks towards the core network:

- Mobility management
- Session management
- Identity management
- Call control

### 3.3.3 eNB (base station) in LTE

In GSM networks this task can be done with base transceiver station (BTS). eNB is the hardware which can be connect with mobile phone to communicate directly with UEs through wireless infrastructure. The eNB stands for evolved NodeB. This can be

often abbreviated as eNodeB or eNB. In LTE architecture, the eNB is consider as a complex base station whose basic task is to handles the radio communication with multiple devices in the cell. The eNB can also be used to carries out the radio resource management. LTE model do not consists of any radio resource controller. However, Node B can be considered as a minimum functionality and can be controlled by radio network controller. In this way, it can simplify the architecture and allows the less response times.

## 3.4 Security analysis

The DM-GKM security is based on the assumptions of finding the set of pairwise relatively prime numbers, and an equation modulo of their product. The proposed framework also provides other security attributes and resilience against all possible attacks such as man-in-the-middle, replay attacks prevention. These security attributes assure the secrecy of the system and usually called the goals of the security system. The common security requirements that a secure GKM framework must satisfy are forward/backward secrecy, authentication, data integrity, confidentiality, integrity, anonymity, non-repudiation, and replay attacks prevention. In Chapter 5, we performed the security analysis which indicates that the DM-GKM consider various security requirements for group based applications. Moreover, the security analysis is also done for proposed DM-GKM by using the BAN logic and Markov chain model. The following sections present the detail descriptions of these two methods.

### 3.4.1 BAN logic

BAN logic name given by its inventors, Mike Burrows, Martıın Abadi, and Roger Needham (Kyntaja, 1995). This logic is stated as the logic of belief and action. This consists of a set of rules in order to describe and examine the information exchange for

a protocol (Burrows, Abadi, & Needham, 1989). The main use of BAN logic is to examine the authentication protocols by developing the beliefs that truthful principles executing correctly as a result of protocol execution. As an example, Alice might come to believe that the key which is she using is the good key which she is using to communicate with the Bob. The "idealized" form of the protocol can be obtained by using the logical formulas. As an example, if a server sends a message to Alice with session key which is hidden inside the session key, this key can be replaced by a formula that means the key is a good key. The inference rules are then be written which are based on Alice's capacity to decrypt the key and other assumptions. The motivation behind the BAN logic is to make the mathematician's credo to make the needed distinctions (Nessett, 1990). The main concept is to make said the goodness of the keys. Let's suppose the following scenario:

- The proposition 'P' is believed by Alice, it can be written as: $A| \equiv B$ and say 'A believes P'.
- Alice believes that the key $K_{AT}$ is a good key for communicating with Trent. We say A believes $K_{AT}$ is a good key for A and T This is expressed as $A| \equiv A \overset{K_{AT}}{\longleftrightarrow} T$.
- Trent can be believed as an authentication server or certification authority. If Alice considers that Trent can be trustworthy to generate a 'good key' for communication with Bob, this scenario can be expressed as: $A| \equiv T \Rightarrow A \overset{K}{\leftrightarrow} B$, we say 'A believes T has jurisdiction over or a good keys for A and B'.
- When Alice receives a message this can be written as: $A \triangleleft P$ and say A sees P.
- By seeing a messages does not believe that unless it can be known that who said it. This implies that if Alice sent a message consisting the information of statement P, this can be written as: $A \mid\sim P$ and say A once said P.

- It can be believe that Alice's statement is fresh. When a statement P is fresh we write #(P) and say P is fresh.

### 3.4.2 Markov chain model

Markov chain model is used to derive the analytically average update cost of state transition. The rekeying cost such as join/ leave and switch operations is calculated and analyzed by Markov model (Gilks, Richardson, & Spiegelhalter, 1995). The proposed framework consider the arrivals of new group member followed the Poisson process (Chang & Kuo, 2009). The duration of the member stays in the group session is distributed by random variable. This analytical model is used to calculate the cost of the state transition. The arrival rate (arrivals/time unit) of the member in the group can be considered as the poison process. The duration of the members within the group is represented as the exponential distribution with mean duration of 1/μ time units. The average number of the users in the SG can be considered as the μ. The parameters $\lambda$ and $\mu$ can be changed over time by the SG controller. This can adjust the estimation of $\mu$ every time unit to better approximate the value of rekeying overhead. Therefore, the value of rekey can be then modeled by using the Markov process (Monahan, 1982; Yu, Tang, Mason, & Wang, 2010).

### 3.4.3 Statistical Analysis

To determine the relation between different variable of proposed framework in order to verify the results, the Statistical Package of Social Sciences (SPSS) 24.0 (IBM, 2017) for data analysis is used. The Normality test is performed to check the normal distribution of the data. The results from the two continuous variables "storage overhead" and "no of encryption" are taken to check the normality of this data set.

Moreover, the regression analysis is also design for estimating the relationships among different variables. Detail analyses of these methods are presented in Chapter 5.

### 3.5 Results gathering and comparisons

The implementation part helps us to analyses the proposed framework against different schemes. In the resource constrained environment of wireless mobile networks, multiples factors such as highly dynamic environment, high number of membership's changes, and bandwidth constraints must be considered for efficient and successful multicast communication. The performance analysis is designed to check the correctness and effectiveness of the framework. These analyses help us to define and check the systems in different scenarios such as storage, communication and computational overhead. These all results are gathered and analyzed against well-known approaches to check the proposed framework efficiency and suitability for group based applications. Through performance analysis and simulations, we demonstrate that DM-GKM framework provides the lightweight key management framework for large and dynamic groups by considering less storage and fewer computations of keys.

### 3.6 Chapter Summary

In this Chapter, the methodology of research including approaches and key steps are described to specify the required guidelines for the development of GKM framework in wireless mobile environments. The explanation of well-known advance LTE technology is also explored with detail that show how different components are treated in the simulation environment. Next Chapter looks at the provision of key management framework for group based applications in wireless mobile environment.

# CHAPTER 4: PROPOSED GROUP KEY MANAGEMENT FRAMEWORK FOR GROUP BASED APPLICATIONS

The aim of this study is to propose a lightweight key management framework for dynamic mobile members in group based applications. A new, improved key management framework is presented in this Chapter, which is given the name "DynaMic Group Key Management" (DM-GKM). The DM-GKM is based on the application of an asymmetric encryption scheme in a decentralized architecture with independent key for each cluster in order to improve the rekeying performance by minimizing the rekeying overhead.

GKMPs are generally based on collaboration to establish a secure communication network between the members and service providers. For secure group based communications, the common method is to use symmetric keys which are shared by all group members and used to encrypt the transmitted data. Newly joined members should be restricted by accessing the previous content and the leaving members from accessing future content, the keys must be refreshed after each membership change. This can be achieved easily by allowing the GC to share a unique key which is called the Key Encryption Key (KEK) with every member. When there is a membership change, the GC uses the individual KEK of every member to encrypt the new TEK. This is secure but inefficient method because the cost of the TEK updates grows linearly with an increase in group size. Therefore, the rekeying process becomes a critical problem in multicast key management with multiple handoffs (Wallner, Harder, & Agee, 1999).

In the simplest arrangement, all the legitimate group members share a secret group key through the TEK, which is assigned by the GM. Each individual member receives a new TEK encrypted by the GM; rekeying messages are constructed including the

encrypted content. Rekeying operation is done by multicasting all the rekeying contents and each legitimate group member only performs a one-time decryption. Even though these schemes are appealing because of their simplicity, they are not scalable with extensive latency and GM saturation (Harney & Muckenhirn, 1997) due to high mobility in dynamic group based applications. In addition, the frequent movement of members from one SG to another within the group results in system bottlenecks. Having a secure way of multicasting messages to group members in wireless mobile environment guarantees access control mechanism, which ensures:

- Confidentiality
- Integrity
- Security
- Backward and forward secrecy

Efficient key management operation is challenging due to the dynamic nature of the group members because of frequent joins and leaves and switch operation, which will trigger the rekeying process. In this situation, the key server generates and delivers the new TEK in order to invalidate the old TEK. This process ensures that future/previous messages after the member joins/leaves fulfil forward/backward secrecy. In this way, the load is increased on the key servers in order to support multi-group host mobility services (Wong et al., 2002; Han et al., 2011; Je et al., 2010; Srinivasan et al., 2010). GKMPs have been studied extensively over the years in order to reduce the rekeying overhead (Canetti et al., 1999); (Harney & Harder, 1999).

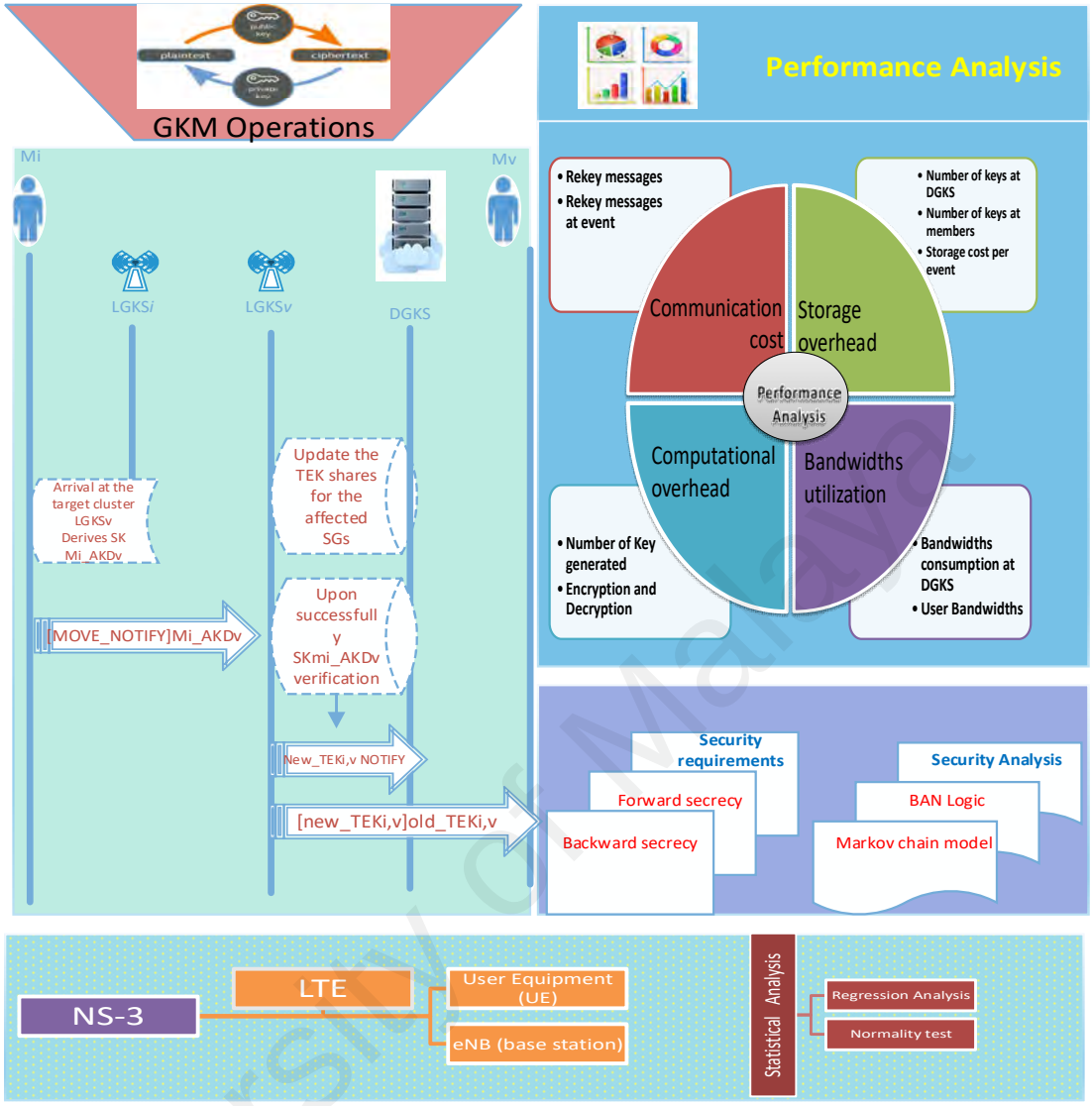


**Figure 4.1:** Proposed DM-GKM Framework

A new key management framework as shown in Figure 4.1 for group based applications is proposed in this study, taking into account the host mobility scenario of the group members. This Figure illustrates the detail representation of proposed framework model that includes different steps of lightweight group key management framework.

The RSA public key cryptography algorithm is used in this study to determine the effectiveness of the proposed framework whereas the CRT algorithm is used to

accelerate the decryption process for multiple dynamic members. The RSA-CRT algorithms are presented in more detail in the following sections. A new key generation and key modification algorithm is developed to account for the host mobility scenario, whereby the members join or leave the group, or move from one SG to another within the group. The DM-GKM is designed in order to provide users with a well-organized, lightweight key management, which offers fewer computations of keys, null rekeying mechanism for host mobility and secure forward and backward secrecy for the group members. One of the most challenging tasks in the development of key management is managing the keys for group based applications in a highly dynamic wireless mobile environment.

This Chapter is organized as follows; in Section 4.1, the RSA and CRT algorithms are presented. The section also discuss the way that how to use the RSA and CRT algorithms to accelerate the key management process. Afterwards, Section 4.2 presents the proposed framework along with the detail design and operation of key management. The network model and initial setup are presented in Section 4.3. Section 4.4 presents the rekeying scenario of membership change in DM-GKM. Section 4.5 outlines the discussion of this Chapter. Finally the Chapter ends with a summary in Section 4.6.

## 4.1 The adoption of RSA and CRT in proposed solution

As studied in Chapter 2 the use of symmetric keys, the schemes may introduces rekeying overhead due to the dynamic nature of the mobile members which considered large number of TEKs and KEKs to update the keys across multiple groups. By considering these limitations, the proposed framework is aims to reduce the heavy rekeying overhead which is frequently caused by the key symmetry i.e. one identical key is shared among group members. By removing this shared key symmetry, we

introduced the DM-GKM framework which aims to decrypt the ciphertext of same plain text with different keys. Moreover, other keys still remain valid if revoking one user's private key. The DM-GKM is based on RSA and CRT based public key cryptosystem, in which every member in the RSA algorithm has a key pair of public and private keys. Thus, act as an encryption/ decryption in an asymmetric pairwise manner. The idea is to use the asymmetric cryptography keys to distribute the group key and user private keys to all the group members. The following sections present the RSA and CRT algorithms in detail.

### 4.1.1 RSA algorithm

RSA algorithms can be used for public key encryption and digital signatures. Public key cryptography is also known as the two key cryptography because it involves two keys; public key, which may be known to everyone in the group, and user secret key or private key which is only known to legitimate user. The public key is used to encrypt the message and for verification of the signature. Users use their private key to decrypt the messages and to create the signature. The RSA security depends on the factorization of the greater number which makes the factorization of a coprime factor using that number. Therefore, the difficulty of RSA is depends on the factorization on the numbers (Nitaj; Van Tilborg & Jajodia, 2014). Therefore, the implementation of RSA is more difficult and time consuming for resource constrained wireless mobile environment (Rivest et al., 1978). As an example a member who is sending the message contains the prime numbers $p$ and $q$ such that $p \neq q$ and opens the product of $n = p \times q$ is called the RSA modulus which is known as the public key to the public. To attain the sufficient security, the values of p and q must have a length of at least 512 bits. The sender chooses a number $e$, which have a relatively prime relationship to Euler quotient function $\varphi(n) = (p-1)(q-1)$. Then the value of $d$ can be calculated such that $e \times$

$d = 1 mod \varphi(n)$ as the private key (Barrett, 1986). The detail steps of RSA cryptosystem are as follows:

1. Randomly choose two large primes. Generates two different primes' $p$ and $q$. Let $p$ and $q$ be very two large primes of nearly the same size.

The $p$ and $q$ should be co-prime and individually prime as well. When the $p$ and $q$ are individually prime, the value Euler's totient of $n$ into the product of totient of $p$ and $q$, such as:

2. Calculate the RSA modulus $n$ with two primes where
   i. $n = p \times q$
3. Calculate the totient $\varphi(\mathrm{n}) = (p-1)(q-1)$
4. Select the public exponent the integer e such that
   ii. *1 < e < $\varphi(n)$ and gcd ($\varphi$ (n), e) =1*
   iii. This is the number of positive integers less than $n$ and relatively prime to $n$
5. Calculate for the private exponent $a$ value for $d$ such that
6. *d = e −1 mod $\varphi$ (n)*
   iv. Finally, use the extended Euclidean algorithm to compute a unique integer $d$, such that
7. $e \times d \equiv 1 mod \varphi(n))$
8. Public key = *[e, n]*
9. Private key = *[d, n]*

Note that the main source of security in RSA is keeping $p$ and $q$ secret and therefore also keeping $\varphi(n)$ secret. The flow chart of RSA algorithm is shown in Figure 4.2.

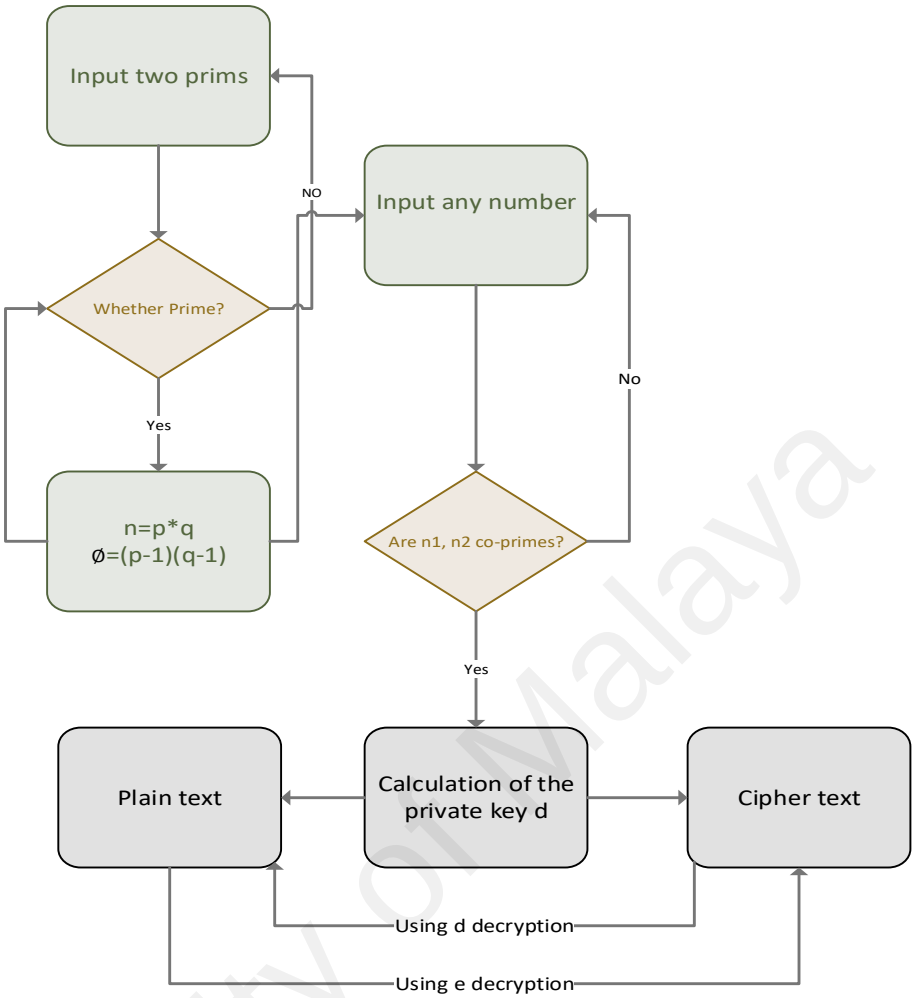


**Figure 4.2:** Flow chart of RSA algorithm

As an example, when a source node wants to encrypt message for sending it to destination node, it uses public key of destination node using RSA in following way:

$$C = (M^e mod\ N) \quad \text{----------} \quad 1)$$

where, C=ciphertext and M=plaintext

The decryption is performed using

$$M = (C^d mod\ N) \quad \text{----------} \quad 2)$$

Encryption and decryption are carried out using two different keys. The two keys in such a key pair are referred to as the public key and the private key.

### 4.1.2 Chinese Remainder Theorem

A Chinese scientist Sun-Tsu around A.D 100 solved the problem of those integers '*x*', that gives the remainders of 2, 3 and 2 when divided by 3, 5 and 7 respectively. One of the solutions is 23, which can be assumed that all the solutions are comes from 23+105k, where '*k*' is the arbitrary integers. It estimates a correspondence between systems of equation of modulo. That implies that a set of pairwise relative prime moduli and an equation modulo of their product. Let $n1, n2, n3 \dots nk$ be pairwise relatively prime integers i.e. $\gcd(n_i, n_j) = 1$ when$i \neq j$. Furthermore, If $a1, a2, a3 \dots ak$ are any integers, then there exists a unique integer *x* modulo $M = n1, n2, n3 \dots nk$ that satisfies the system of linear congruence. Let us look at a simple interpretation of the theorem:

$$X = a1 \ (mod\ n1),\ X = a2 \ (mod\ n2), X = a3 \ (mod\ n3) \dots X = ak \ (mod\ nk)$$

Moreover,

$$X = a1M1y1 + a2M2y2 + \cdots akMkyk \ (mod\ M) where\ M_i = \frac{M}{n_i}\ and\ M_i y_i$$

$$= 1(modn_i)\ for\ i = 1,2, \dots k.$$

*or*

$$X = \sum_{i=1}^{k} a_i M_i y_i \ (mod\ m) \qquad \text{----------} \quad 3)$$

*where*

$$M = m1, m2, \dots mk$$

*and*

$$M1 = \frac{M}{m1}, \ldots Mk = \frac{M}{mk}$$

$$y1 = M^{-1}\, mod\ (m1), yk = (Mk)^{-1}(mod\, mk)$$

### 4.1.3 Combining RSA and CRT

In this study, the CRT is used for efficient decryption method in RSA cryptosystem. The decryption process becomes accelerate when combining CRT with RSA method. This scenario can reduce the computational cost and rekeying overhead in resource constrained mobile devices as compared to traditional methods (Ding, Pei, & Salomaa, 1996).

The complexity of the RSA methods depends on the size of the decryption exponent '*d*' and on the modulus '*n*'. This decryption exponent defines the number of modular multiplication required for exponentiation. However, the modular '*n*' determines the size of the intermediate results. One of the idea to reduce the size of '*d*' and '*n*' is by using the CRT (Quisquater & Couvreur, 1982). Thus, for fast decryption of RSA the CRT can be applied during generation of private key and decryption. The combinations of RSA-CRT methods are differ from the standard RSA in key generation and decryption process. In this method, the public key consists of the modulus '*n*' and private exponent '*d*' which should be kept secret. The value of this private exponent '*d*' cannot be made short because the RSA system can easily be broken for the smaller values of '*d*'. The RSA modulus is divided into two independent sub-modulus, these independent modulus can be used for both encryption-decryption processes (Ding et al., 1996). The CRT version of decryption might be considered as an extra source of insecurity because it requires the primes numbers '*p*' and '*q*' and the decryption

exponent *d*. However, the simplest way to find a factor of modulus 'n' gives the decryption exponent '*d*', therefore, no security lost by using this method. Most of the researcher use the CRT to computes the RSA signature, because the primes of RSA-CRT are four times faster than computing 1024 bit RSA. In this scheme, the primes become smaller as compared to traditional RSA method. As an example, by using the three primes (each 341 bits), the performance of RSA-CRT becomes nine times faster as compared to 1024 bit RSA while achieving same level of security (Zheng, Huang, et al., 2007).

When using RSA with CRT, the values of $p$ and $q$ are precomputed and stored as the public key. The values of $p$ and $q$ are known to private key owner. In this arrangement, the public key is the same $(e, N)$ as traditional RSA, however, the private key now becomes the tuple of $(d_p, d_q, p, q)$ $where$

$$d_p = d(mod(p-1) \qquad \text{---------- 4)}$$

$$and$$

$$d_q = d\big(mod(q-1)\big) \qquad \text{----------5)}$$

Because of the Chinese Remainder Theorem (and because p and q are relatively primes), the value of $m$ can be deduced immediately as: $m = (M^d mod\ N)\ mod\ pq$ which is exactly what we were trying to compute. This process is fairly simple to see the correctness of the algorithm. Moreover, to achieve the sufficient level of security the prime factors of $p$ and $q$ of modulus in RSA methods should be strong primes.

*(a)* ***RSA-CRT key generation algorithm***

The public key is $(N, e)$ and the private key is $(p, q, d_p, d_q)$

1. Suppose $p$ and $q$ are large primes numbers such that $gcd(p-1, q-1) = 1$
2. Compute $n = p * q$ and
3. $\varphi(n) = (p-1)(q-1)$
4. Choose two integers $d_p$ and $d_q$ such that
    v. $\gcd(d_p, p-1)\ and\ \gcd(d_q, q-1) = 1$
    vi. Find $d$ such that
    vii. $d = d_p(mod\ p-1)$
    viii. And
    ix. $d = d_q(mod\ q-1)$
5. compute $e = d^{-1}(mod\ \varphi(n))$

Following steps can be applied for decryption process. lets $M$ is the plaintext and $C$ the cipher text of $M$. If $C$ is not dividable by $p$ and $d_q == d\ mod\ p-1$, then $C_{dp} == Cd\ mod\ p$. For decryption we find $M_q = C_{dp}(mod\ p) = C_d(modq)$. Then using Chinese Remainder Theorem, we can find the solution:

$$M = M_p(mod\ p) = C_d(mod\ p)$$

and

$$M = M_q(mod\ q) = C_d(mod\ q)$$

Hence $M$ can be obtained as:

$$M = C_d mod\ N \quad \text{---------- 6)}$$

The following section presents the key generation and distribution of the RSA-CRT in DM-GKM. The section also presents how the keys are changed and modified upon join, leave and switch operations.

## 4.2 DM-GKM: Proposed Group Key Management Framework

The proposed framework aims to support secure group based application in wireless mobile environments. The proposed framework is the two tier decentralized framework as presented in (Abdmeziem, Tandjaoui, & Romdhani, 2015; Kiah & Martin, 2007). The framework model is shown in Figure 4.1. In this framework model, the first tier is the domain level which consists of the core part of the wired network, and the *DGKS* for key generation and authentication procedures of group members. The second level is the wireless part which consists of clusters and have the area key distributor managed by multiple areas key servers *AGKS*. Each cluster in DMGKM has a wireless LAN with an access router and many access points. In each area there are number of members which belongs to one or more SGs. The proposed framework adopts the independent *TEK* (Kamat et al., 2003; Mittra, 1997) per SG to alleviate the problem of 1-affect-all phenomenon and to localize the rekeying process. This implies that if the rekeying process occurs, it can be handled locally without affecting other SGs. The local rekeying procedure gives *DGKS* and to *AGKS* scalability, track the mobility member and to reduce the need of rekeying when members handoffs while still maintain the group session between different SGs. As can be seen from the Figure 4.1, the *DGKS* is responsible for sending group content to each *AGKS* and storing the main list which is called the $M_mList$. The main list contains mobile members' information regarding join/leave and movement. This list also provides the fast and secure authenticated mechanism for handover members along with initial key establishment. Each *AGKS* is responsible for authentication, generating and sending group key, and finally transmitting content to mobile members.

### 4.3 Initial Setup and distribution of keys

The proposed framework is the combination of RSA based public key cryptosystem and Chinese Remainder Theorem. In this way, each member in the system has a pair of public and private keys. These keys act as an encryption and decryption in an asymmetric pairwise manner. The DK-GKM also adopts the independent group key for each cluster. Thus, a key pair can be changed by just modifying the public key without changing the other user's private keys. Table 4.1 presents the notations and symbols used in this Chapter.

**Table 4.1:** Notations and symbols used in proposed scenario

| Symbol | Descriptions |
|---|---|
| $G^{TEK}$ | Group Key |
| $G^{TEKi}$ ($G^{TEK1}$, $G^{TEK2}$, $G^{TEK3}$) | Independent key for each subgroup. |
| $S^{KEK}$ | Sub group Key |
| *AGKS* | *Area Group Key Server* |
| *DGKS* | *Domain Group Key Server* |
| *TEK* | *Traffic Encryption Key* |
| *KEK* | *Key Encryption Key* |
| ($S^{KEK1}$) of $SG^{1}$ | *Sub-Group Key of group 1* |
| $S^{KEY}$ | *User key* |
| $O_m List$ | Old member list |
| $M_m List$ | Member's mobility list |
| $C_m List$ | Current members list |
| \|\| | Concatenation operator |
| → | Unicast message |
| ⇒ | Multicast messages |

The main key server is represented as the Domain Group Key Server (*DGKS*) and as the entire system is divided into the different areas or groups and each SG is maintains by Area Group Key Server (*AGKS*). Each SG has its own sub group Server (Local Group Key Server) and maintains the sub group key. Figure 4.3 shows the reference framework for key generation and key distribution scenario for group based applications. The DGKS maintains the main mobility list called the $M_mList$ and the LGKS maintain the list of current members $C_mList$ as well as old members which is called the $O_mList$.

As shown in Figure 4.3, wireless network is divided into different areas. Each *AGKS* is responsible for connecting mobile members to the network. In wireless network level, mobile members switch between areas while maintaining the group session. Each member in the network belongs to one or more sub-groups in the total of $SG_m$ denoted by ($SG_1$, $SG_2$….$SG_m$).

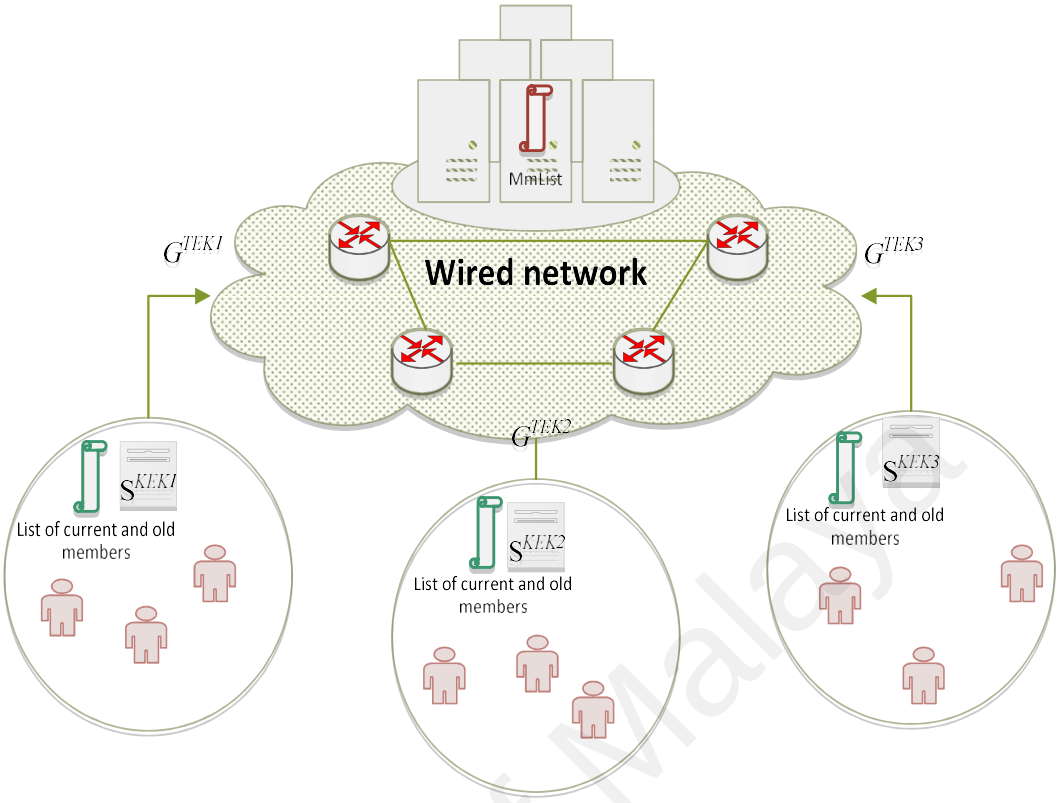


**Figure 4.3:** Reference Framework

In the proposed scenario, keys can be classified into three types according to their roles in the system, i.e. the public key or the group key ($G^{TEK}$), Sub-group Key ($SG^{KEK}$), and Individual Key or secret key ($S^{KEY}$) of members. The *TEK* is on the top level of the key and is used to encrypt the multicast data, and to distribute the public key to the subgroups. The KEK is used to distribute the TEK in each subgroup and private key to the members. The main list is called the $M_mList$ stored in the main server supports members topology control and member authentication information. Authentication information of registered members is saved in this list globally. Different *AGKS* refer to this list to recognize authorized group members' location or their join/leave or handoffs information. The information in the main list enables servers to have topology control on different mobile groups because all the information about members' behavior, join/leave and movement are necessary to support group based applications. Moreover,

each *AGKS* keeps two lists: first list is used for current members (denoted by $C_mList$), which contains identities of current members of the group; and second list is used to contain the information of old members (denoted $O_mList$) which were members of area $AGKS_i$ and moved to other areas without leaving the secure group session. The detail steps of TEK and KEK keys generations and distribution are shown in Figure 4.4.

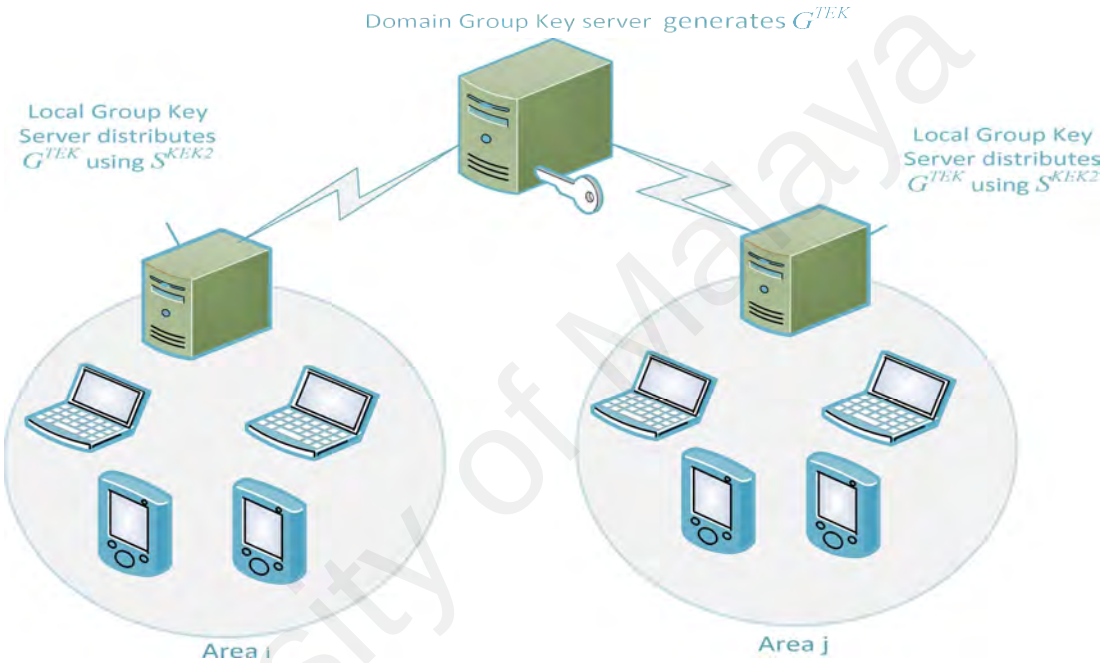


**Figure 4.4:** Generation of Keys of DGKS and LGKS

### 4.3.1 Key distribution operations

The *DGKS* is responsible of managing the key related task as shown in figure 4.3 and in 4.4. The following steps are done for the initialization of the group setup.

***Step 1:*** Then *DGKS* derives the necessary cryptography keys i.e. as many as public-private keys pairs as the number of $SG_s$ to setup the group by using the proposed algorithms. To accomplish this task, it is to be assumed that the DGKS has a large

number of pairwise relatively prime numbers. Since these prime numbers need to be provided by the DGKS when needed in the case of key generation and modification

***Step 2:*** The $G^{TEK}$ is used by the *DGKS* to delivers the multiple subgroup keys $SG^{KEK}$. Because the distribution of a new $G^{TEK}$ within an area must itself be secure, the $SG^{KEK}$ is used to encrypt the transmission/distribution of the $G^{TEK}$ within each $SG_s$. The leaf nodes are the member's nodes and contain the $S^{KEY}$ of all the members within $SG_s$.

***Step 3:*** *DGKS* also generates the $M_mList$ which consists of the registered members in each area. The member in the system has key pair that contains the public and private keys, in which each key can be used to encrypt and decrypt the message in an asymmetric pairwise manner. Each private/ secret key $S^{KEY}$ is used by each member to decrypt the key information sent by the *DGKS*. The *DGKS* completes the initial setup by generating and distributing all the keys to the members in $SG_s$.

For example, the Figure 4.3 shows the concept of proposed framework. In this Figure the basic idea is to establish a public key, which is called the $G^{TEK}$ in our system. The $G^{TEK}$ encrypt the content which can be decrypted by different private keys. The importance of this setup is that one of the key pairs can be modified by only altering the $G^{TEK}$ without changing the other member's key pairs. In this scheme the rekeying cost of the system can be alleviated by using the asymmetry keys.

## 4.4 Rekeying at membership change

In the DM-GKM, the membership change of a member can be considered as the moving from one SG to another SG. The section presents the detail scenario of membership join, leave and switch operations.

### 4.4.1 Join rekeying

In DM-GKM, after finishing the initial group set up, suppose a member $M_4$ wants to join the SG-1. $M_4$ broadcasts a join-request message to *DGKS*. Upon receiving the message, it will first go through the same authentication process as other members goes to join the group session. Upon verifying the legitimacy, *DGKS* establish a new $G^{TEK}$, and the new $SG^{TEK}$ encrypted with new secret key $S^{KEY}$ of $M_4$, and distributed to this new member and to the current group members. *AGKS* adds $M_4$ to its list of area member $C_mList$.

DGKS can update the new keys by the following message to the new joining member:

$$DGKS \rightarrow AGKS_1 \rightarrow \{distribute||E(G^{TEKNEW}, SG^{KEKNEW})_{SKi}\}, i = 4$$

And multicast the new group key to the other members in its area, encrypted with old TEK.

$$DGKS \rightarrow AGKS_1 \Rightarrow M_1, M_2, M_3, M_5 : \{update||E(G^{TEKNEW}, SG^{KEKNEW})_{GTEKOLD}\}$$

Note that, the other SGs will not effect from this joining member as the framework is using the independent TEK for each SG. Hence, giving the whole system scalability and mobility dynamism. The multicast message is delivered to the area where a new member joins and an additional single unicast message to a newly join member. As a result, the intruder can also not be able to access the group because the intruder does not have the $G^{TEK}$. The newly join member do not have the old TEK, therefore, cannot decrypt the other contents of the group. The *backward secrecy* should be guarantee as the *AGKSi* should commit to the new $G^{TEK}$ and distributes it to all the members of its cluster. This scenario is illustrated in Figure 4.5.

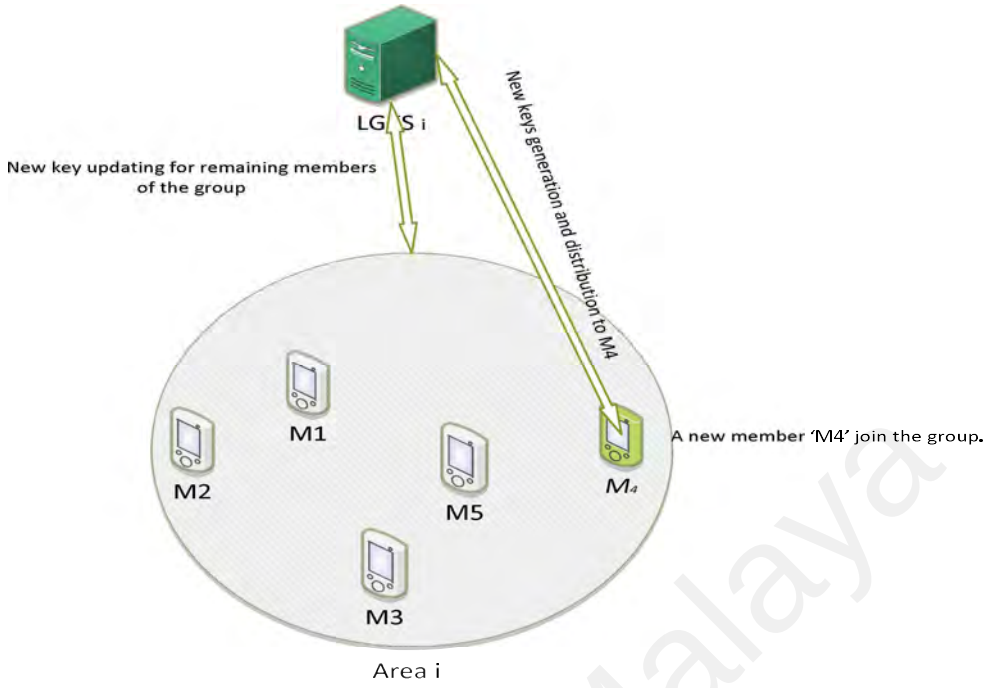


**Figure 4.5:** Join protocol scenario

### 4.4.2 Leave re-keying

Upon leaving a group, a new $G^{TEK}$ is generated and distributed only to the remaining members. This scenario is illustrated in Figure 4.6. A valid member located in $SG_i$ and wants to leave the $SG_i$ by sending a leave signal to the $AGKS_i$. In this case, all the $AGKS_s$ in which this member visits should commit to a new $G^{TEK}$ and distribute it to the members in their clusters, in order to guarantee forward secrecy. For example, in Figure 4.6, a member $M_7$ leaves the group. Before transmitting the data, *DGKS* must update the keys from the following equation:

$$DGKS \rightarrow AGKS_1 \Rightarrow M_6, M_8, M_9, M_{10}: \{update || E(G^{TEK}, S^{KEY}{}_6, S^{KEY}{}_{8,} S^{KEY}{}_9, S^{KEY}{}_{10})\}$$

Members in $SG_i$ can update the new keys by this message. Furthermore, forward secrecy can be achieved as the private key of the member not assign to other members

of the group. After distributing all the keys and expelling the members from $AGKS_j$, the $C_mList$ of $AGKS_i$ should be updated.

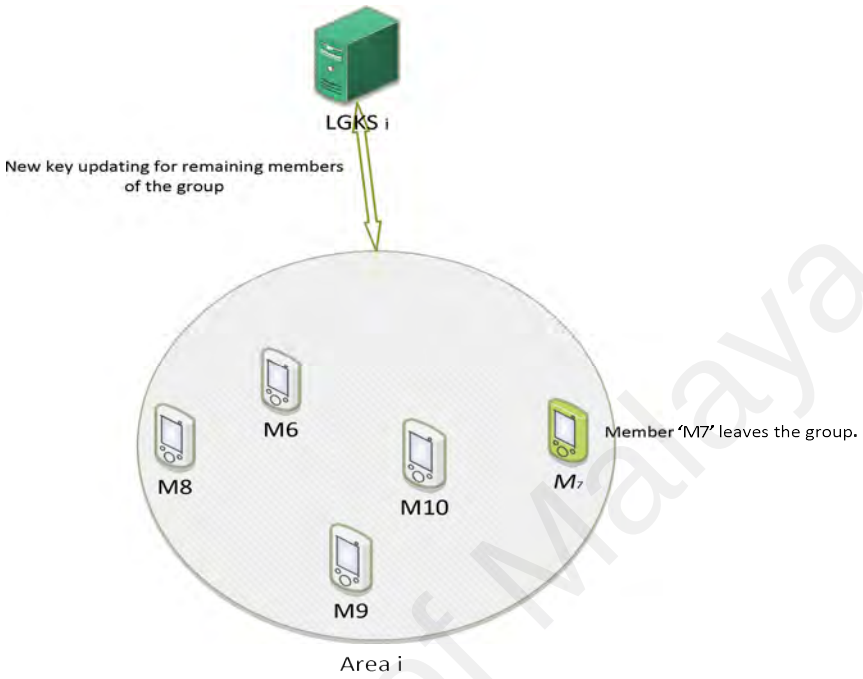


**Figure 4.6:** Leave protocol scenario

### 4.4.3 Rekeying for switching members

Consider the scenario where member $M_3$ belongs to $SG_i$ as shown in Figure 4.7 and wants to moves to $SG_v$ and leaves the $SG_i$ without leaving the network session. The *DGKS* should switch $M_3$ membership from $AGKS_i$ to $AGKS_v$. In this case the $M_3$ loses the membership of $SG_i$ but gain the membership of $SG_v$.

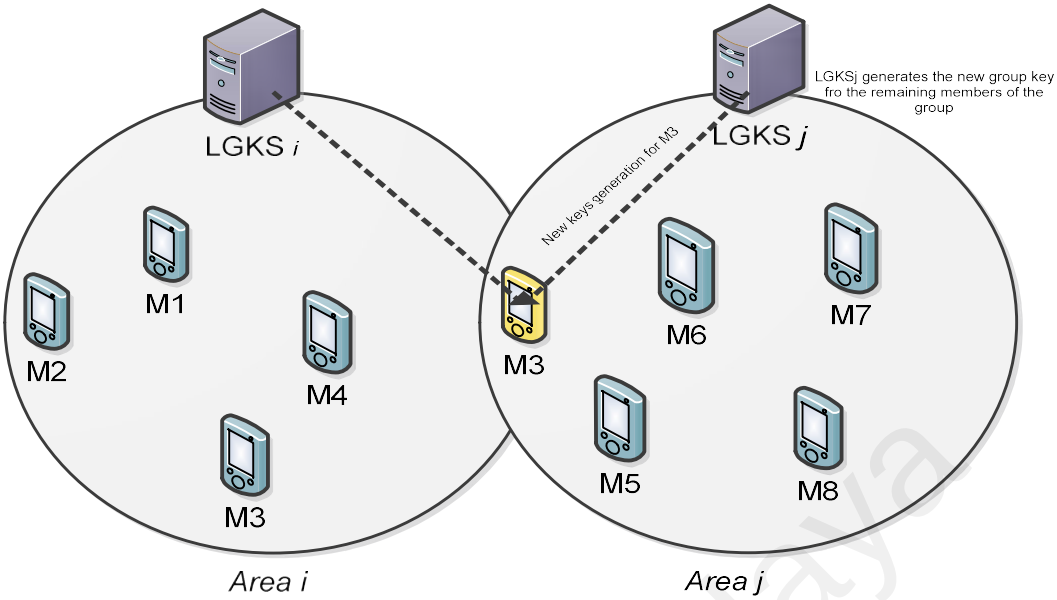


**Figure 4.7:** Membership movements within group

For security purposes, $AGKS_v$ need to verify the the $M_3$ is a valid member and really comes from area $AGKS_i$. For this purpose, when $M_3$ moves from area $AGKS_i$ to area $AGKS_v$ the following operations are then executed:

***Step 1:*** $M_3$ simultaneously sends a move_notify message to both $AGKS_i$ and the target $AGKS_v$ encrypted under its secret key $S^{KEY(AGKSi)}$ to $AGKS_v$ i.e.

$$M_3 \Rightarrow AGKS_i, AGKS_v: \{move_{notify}\}_{S^{KEY}(AGKSi)}$$

***Step 2:*** When "move_notify" message received by $AGKS_v$, it verifies that $M_3$ is a legitimate member and really comes from $AGKS_i$ by using the already secret key of $M_3$ stored in the $M_mList\ of$ DGKS. If the verification succeeds, $AGKS_v$ adds the visitor $M_3$ to its list of members $C_mList_v$.

***Step 3:*** Upon joining the $SG_v$ the $TEK_v$ should be updated by *DGKS* to meet the requirements of the backward secrecy. This corresponds to the rekeying process inside $SG_j$. Since $G^{TEKj}$ which has been shared by group of users in $SG_j$ would expire, the

*DGKS* conveys the new $G^{TEKjnew}$ to the members, through a set of multicast rekeying messages as:

$$DGKS \rightarrow AGKS_j \Rightarrow M_1, M_2, M_4 : \{update || E(G^{TEK2New})\}$$

***Step 4:*** The $AGKS_i$ put $M_3$ to its $O_m list$.

***Step 5:*** As we can see, the requirement of *backward secrecy* is also meet as the $M_3$ cannot access the previous data of $SG_v$ as the member cannot be able to access the data before its join because the member don't have the $G^{TEKv}$ before its joins. By this arrangement, the data which is encrypted with the new $G^{TEKv}$ can still be decrypted by all the members by using their secret keys ($S^{KEY5}$, $S^{KEY6}$, $S^{KEY7}$, *and* $S^{KEY8}$). Therefore, it guarantees the backward secrecy requirement of DM-GKM framework. Note that each *LGKS* only contain the information of existing members which are currently serving in the group in order to enhance the storage capacity. Figure 4.8 presents the detail flowchart of member join, leave and switch operations.

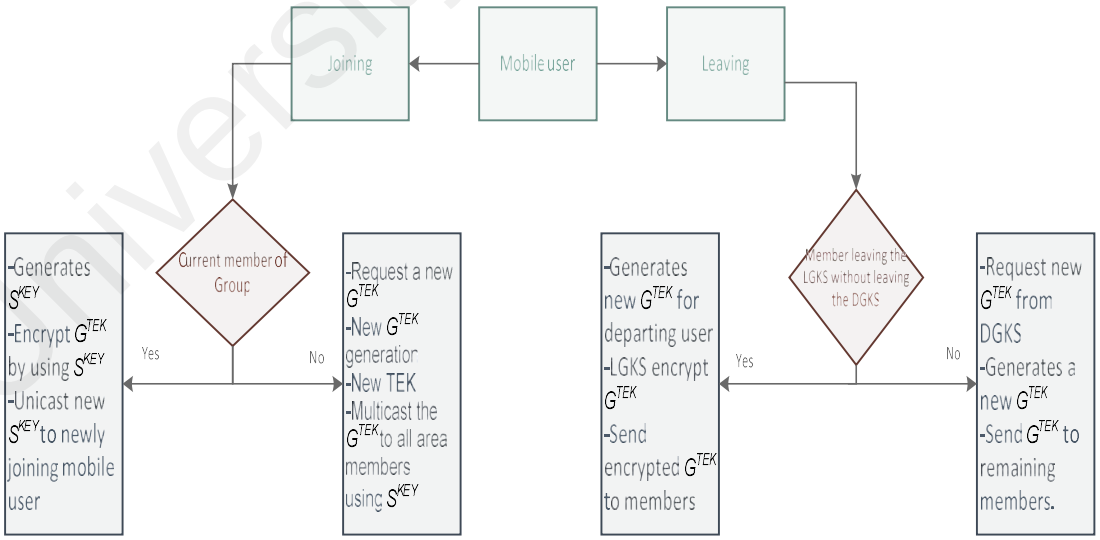


**Figure 4.8:** Flow chart of the member join, leave and mobility scenario

## 4.5 Discussion of proposed DM-GKM framework

The key management framework proposed in this study alleviates the rekeying overhead for backward and forward secrecy during joining, leaving and switching operations. In wireless mobile networks, the number of mobile members may escalate significantly and the mobile members may move very frequently. The 1-affects-n phenomenon may become severe in this case since the membership of the group members changes numerous times across the networks. When one mobile member moves from one location to another, the member is perceived as switching from one sub-group to another, and the keys need to be updated accordingly while maintaining the group communication session. Hence, it is imperative to minimize the 1-affects-n phenomenon. This phenomenon is represented as a number of rekeying messages in this study. The member rekeying is only consider in the target cluster to achieve the backward secrecy since the member of the previous cluster is listed as valid member in the current session. In this way, the DM-GKM framework significantly reduces key generation and key encryption overhead resulting from join/leave and host mobility scenario compared to other host mobility protocols. The DM-GKM framework also significantly reduces unicast and multicast communication overhead when members join the group. These properties are indeed essential for secure multicast in wireless networks. In wireless mobile networks, in addition to join or leave, the inter-area and intra-group movement of a mobile member is considered as leaving the old cluster and joining a new cluster. The simulation is performed and results are taken to analyses the security requirements for evaluation of the DM-GKM. In addition, the three performance parameters (*i.e.* computational, storage and communication overheads) are also discussed in the next Chapter. The DM-GKM framework offers the following features for group based applications in wireless mobile environments:

a) Effectively maintains a lightweight key management system in wireless mobile environments, which is challenging due to the dynamic nature of the group membership caused by frequent joins or leaves of the members.
b) Ensures backward confidentiality when the mobile members dynamically perform handoff while seamlessly maintain the group communication session.
c) Efficiently manages cryptographic keys for large, dynamically changing group members by means of a decentralized architecture.
d) Reduces signaling load on the main server and provides a separate key for each cluster to prevent the 1-affects-*n* phenomenon.

## 4.6 Chapter Summary

This Chapter presented the detail description of the proposed framework. The decentralized framework is taken for proposed solution for lightweight key management framework. This framework provides the efficient way to generate and distributes the key for dynamic members in group based applications. The study of RSA and CRT algorithms is also presented in this Chapter. This study also explores the way that how RSA and CRT algorithms are combined in order to take the advantages for group based applications. In addition with member's join and leave operations, the proposed framework also established the way to handle the mobility of group members in different clusters. Next Chapter explains the implementation results and performance analysis of the proposed framework.

# CHAPTER 5: EVALUATION OF DM-GKM FRAMEWORK

In this Chapter, the simulation results, performance and security analysis of the DM-GKM framework are presented. The simulation is done to evaluate the performance metrics of the proposed framework. In dynamic group based applications, the rekeying overheads caused by frequent join, leave or move of the group member is the core concerning metric to evaluate the performance of GKM frameworks. The rekeying overhead occurs due to member's movement within the group while maintaining the group session. As a result, storage, computational and communication overheads continuously changes over time due to members location changes dynamically.

The comparison of communication, computation and storage overhead are analyzed with most well-known group key management protocols such as GKMF (Kiah & Martin, 2007), M-Iolus (Mittra, 1997), Decleene models (C. Zhang et al., 2002) and LKH (Wallner et al., 1999) based protocols. We also design our framework by using the Markov chain model to estimate the performance of DM-GKM. The performance of DM-GKM is depends on the number of key generation, key store by key server and members, key encryption and number of transmission to update the keys.

In the following Sections we see the performance overhead with different parameters and compare these with different approaches to check the effectiveness of the proposed framework.

## 5.1 Performance Analysis

In DM-GKM framework, the membership changes are considered as the switching from one subgroup to another. The work analysis the number of rekeying scenario of message transmitted by the DGKS. When a user moves to $SG_j$ from $SG_i$; the DGKS must revoke all the keys that the member of $SG_j$ has shared, this process indicate as the

rekeying process in the area $SG_j$. However in the dynamic group environment, when there are multiple members participate in handoffs in $SG_m$ (where $m$ is the number of subgroups). The convectional protocols experience the huge amount of communication messages and countless delays in obtaining the rekeying messages. If the group key shared with $m$ number of SG then the rekeying updating in all SG becomes the performance hurdle due to *1-affect-n* phenomenon. By observing these limitations, the DM-GKM framework proposed which is very adaptive in multi groups with multi handoffs. As defined earlier, the cryptography keys are used separately for each area in DM-GKM, the rekeying operation is become localized for each SG, hence alleviating the *1-affect-n* phenomenon. Moreover, the DM-GKM also aims to minimize the need of heavy demand of group key of local area like in conventional protocols, hence improving the scalability and security of the overall framework.

In this Section, the formulation of the analytical model of proposed framework which consists of two SGs i.e. $SG_i$ and $SG_v$, where $SG_v$ is the target SG of $SG_i$. Individual $SG_s$ are controlled independently by the corresponding $LGKS_s$. the rekeying overhead caused by the merge of two $SG_s$ together due to member handoff from $SG_i$ managed by $LGKS_i$ to $SG_v$ managed by $LGKS_v$ are formulated analytically by using Markov chain model. When a member switch from a SG to another it is considered as the leaving from a $SG_i$ follows by a join in $SG_v$. This arrangement reduces the complexity for a member join or leave from $O(m)$ to $O(n)$, where $m$ denotes the number of all $SG_s$'s members and $n$ is the number of each SG members. Moreover, the statistical analysis is also used to show the relation between different performance metrics of proposed DM-GKM.

### 5.1.1 Simulation setup

The DM-GKM is simulated in Network Simulator 3.22 simulation software (Carneiro, 2010). The LTE model is used here which is a software library that allows the simulation of LTE networks. It is assumed that each node moves independently with the same average speed. The simulation is run for 0 to 500 nodes to calculate the different parameters of proposed framework. The effect of arriving members in different areas in group based applications is random with percentages varying over time. This simulation has been performed to estimate the overhead corresponding to the additional signaling load caused by different rekeying overhead:

a) The communication overhead has been compared for both unicast and multicast at DGKS, LGKS level and at user level.
b) Storage overhead corresponding to the additional storage capacity of the key management stored by each entity such as members, DGKS and LGKS. Thus, it is the memory capacity needed by each network entity.
c) Computational cost is the complexity of calculating the new key upon membership change.

### 5.1.2 Communication Rekeying Overhead

In this Section, the assessment of the communication overhead of the proposed DM-GKM framework is evaluated. The comparison has been conducted whenever there is the rekeying communication due to the change in membership location. Communication overhead is the combination of the rekeying messages and the signaling load between members and core network. Rekeying transmission messages consists of both unicast and multicast messages. Figure 5.1 demonstrates the transmitted message size with the number of users in the system.

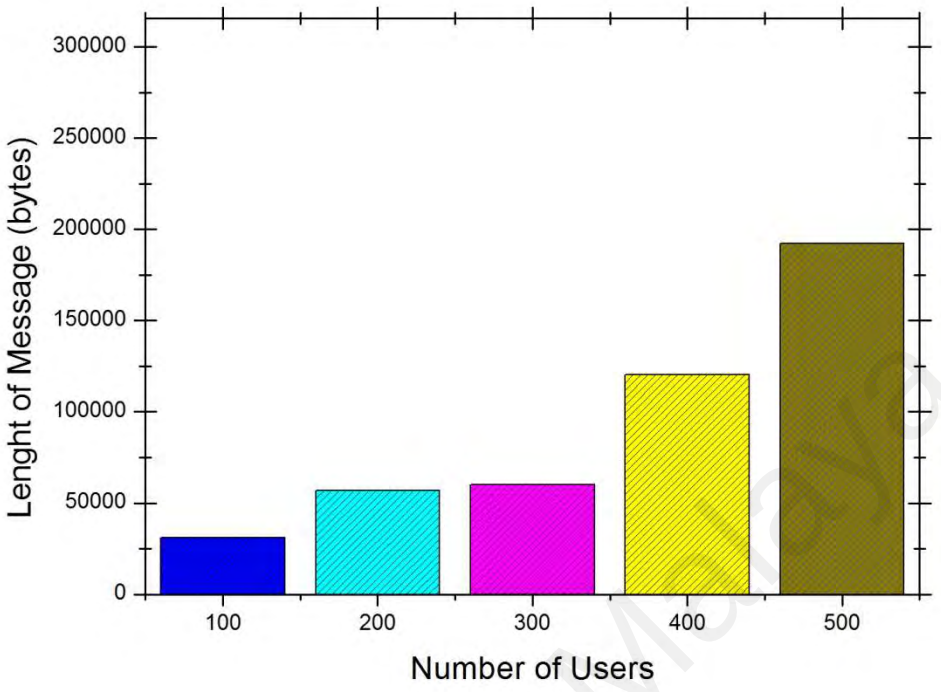


**Figure 5.1:** Size of transmitted message with the varying number of users

Rekeying overhead implies the bandwidths consumptions of resource constrained mobile devices, and other network entities such as LGKS and DGKS. High number of signaling between members and core network can cause a more bandwidth consumption at the LGKS and DGKS. (The detail analysis of bandwidth consumption of different entities in DM-GKM is presented in section 5.1.6) As a result, the fresh key distribution for a new member becomes delay and creates bottleneck. In most approaches when a member leaves a cluster, all the keys should be updated for security purposes. Table 5.1 presents the comparison of signaling transmission between different network entities of DM-GKM with other GKM approaches when member handoffs.

**Table 5.1:** Comparison of rekeying transmission between network entities upon membership change

| Reference framework | (C. Zhang et al., 2002) | (Kellil et al., 2004) | (Kiah & Martin, 2007) | DM-GKM |
|---|---|---|---|---|
| LGKSi↔DGKS | 3 | 1 | 2 | 0 |
| LGKSv↔DGKS | 3 | 1 | 1 | 1 |
| Mi↔LGKSi | 2 | 2 | 2 | 2 |
| Mv↔LGKSv | 3 | 1 | 1 | 1 |

For LKH based approaches, a d-degree tree accommodating '$m$' members is represented as $log_d(m)$. The rekeying overhead for these schemes is represented as $dO(log_d(m))$. On the other hand, the pairwise approaches such as BR (DeCleene et al., 2001), (Kiah & Martin, 2007) introduces more rekeying overhead as the keys at the all clusters and at core network should be updated during membership change.

The methods of GKMF (Kiah & Martin, 2007) and KELLIL et al. (Kellil et al., 2004) introduced the mobility list to track the record of switching members. Thus, the previous cluster also induces null rekeying overhead for leaving member and members maintains the session for visited area. These schemes satisfy the backward secrecy requirement. However, these schemes provide the null rekeying at the cost of storage complexity for the resource limited mobile devices. Moreover, the handover member maintains the local area keys of previously visited clusters to reduce the rekey on return period. Therefore, these approaches induce heavy rekey overhead at each cluster when a member leaves the group after visiting several clusters and creating rekeying overhead in all the visited clusters. As a result, there is very heavy bandwidth consumption for the

whole system. In addition, these schemes remains offline for a longer period of time as until the rekey procedure occurs in all the visited areas.

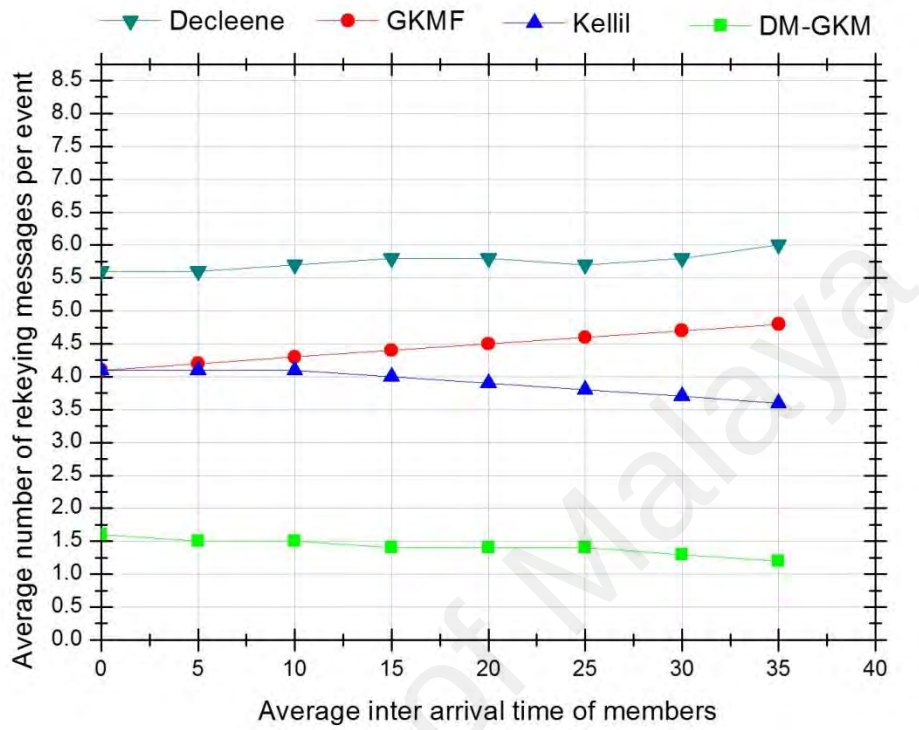


**Figure 5.2:** Impact of membership inter-arrival time on average number of rekeying messages per event

Figure 5.2 demonstrated that the number of rekeying messages in an event is very low as compared to other methods. As DM-GKM considers the moving member from one $SG_i$ to $SG_v$ as the switching member, therefor, the impact of high mobility is very small in proposed DM-GKM framework. This phenomenon also controls the *1-affect-n* with the varying number of inter users arrival. Therefore, the support of high mobility gives better results while reducing the number of rekeying messages.

The analytical model is now conducted to better estimate the rekeying overhead. Let $L_{pi}$ and $J_{pv}$ represent the rekeying messages occurred due to the leave operation in $SG_i$ and join operation in $SG_v$ respectively at steady state probability $p$. The expected

values of $L_{pi}$ and $J_{pv}$ can be denoted as $E[L_{pi}]$ and $E[J_{pv}]$. The analysis has been conducted for a case in which the pool of members arrives according to the certain stochastic process follows the Poisson process with rate λ (arrivals/unit time). Thus the number of concurrent members in a specific SG can be estimated as $n$=λ/μ. The members estimated time in the SG can be mean duration of 1/μ seconds and the service time of the each local area are independent. Let R bits/s represent the data rate for a system. This model can be applied to the DK-GKM with considering independent TEK for each cluster. As defined in (Chan & Chan, 2002), the assumption is that individual parameters of $\lambda$ and $\mu$ are managed independently and the SGs are probable to be joined by moving members. Furthermore, the parameters $\lambda$ and $\mu$ can be adjusted over time due to host mobility scenario, hence, making the LGKS to adjust its estimations of λ and μ every Θ time units in order to estimate better rekeying overhead. Suppose $S = \{0, 1, 2, 3 \ldots\}$ denotes the system state corresponding to the number of concurrent members in a specific SG. Then the system can be formulated using the Markov process (Chan & Chan, 2002) . The $\pi_k$ is evaluated as

$$\pi_p = \frac{n}{p!} \cdot e^{-n} \qquad \ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots \quad 1)$$

where

$$n = \frac{\lambda}{\mu}$$

Hence, the expected number of rekey messages $E[RM_{\lambda,\mu}]$ per unit time can be denoted by using the steady state properties of the Markov chain:

$$E\left[RM_{\lambda,\mu}\right] = \lambda \sum_{p=0}^{\infty} \pi_n \left(E\left[L_{\mathrm{pi}}\right] + E\left[J_{\mathrm{pv}}\right]\right) \ldots\ldots\ldots\ldots\ldots\ldots\ldots 2)$$

In order to simplify the $E[RM_{\lambda,\mu}]$, the approximation value of $\pi_n$ as a $\delta$-function gives the best approximation of

$$\pi_n \approx \delta(p - \frac{\lambda}{\mu}) \quad \text{..... 3)}$$

where

$$\delta\left(p - \frac{\lambda}{\mu}\right) = \begin{cases} 1, & if\ p = \frac{\lambda}{\mu} \\ 0, & otherwise \end{cases}$$

Therefore $E[RM_{\lambda,\mu}]$ can now be written as

$$E[RM_{\lambda,\mu}] = \lambda \sum_{p=0}^{\infty} \delta\left(p\frac{\lambda}{\mu}\right) (E[L_{\mathrm{pi}}] + E[J_{\mathrm{pv}}]) \quad \text{..... 4)}$$

This reduces to

$$E[RM_{\lambda,\mu}] = \lambda(E[L_{di}] + E[J_{dv}]) \quad \text{..... 5)}$$

where

$$d_i = \frac{\lambda_i}{\mu_i}$$

and

$$d_v = \frac{\lambda_v}{\mu_v}$$

Since DM-GKM adopts independent keys per *SG*, the rekeying process only occurred in a specific *SG* where membership change happens, the total rekeying overhead $RO_T$ can be approximated by:

$$RO_T = E\left[RM_{\lambda_i,\mu_i}\right] + E\left[RM_{\lambda_v,\mu_v}\right]) \quad \ldots\ldots 6)$$

The mutual impact overhead by merging the SG$i$ and SG$v$ can be denoted as $E\left[RM_{i,v}\right].E\left[RM_{i,v}\right]$. This explains membership changes from SG$i$ to SG$v$ and their impact on the target SG$v$, and vice-versa. In the first scenario, the member $M_i$ leaves $SG_i$ at rate $\lambda_i$ followed by re-join at $SG_v$ at rate $\lambda_v$. Thus $E\left[RM_{i,v}\right]$ is caused by $E\left[IL_{pi}\right]$ and $E\left[IJ_{pv}\right]$ which are the mutual impact of rekeying overhead at $SG_i$ and $SG_v$ respectively at steady state $p$. By applying the steady state property of Markov process, we obtain $E\left[RM_{i,v}\right]$

$$E\left[RM_{i,v}\right] = \lambda_i E\left[IJ_{pv,}\right] + \lambda_i E\left[IL_{pi,}\right]) \quad \ldots\ldots\ldots\ldots 7)$$

Upon arriving of the member at the target $SG_v$, the $SG_v$ controller $LGKS_v$ multicast the new $TEK_{i,v}$ for the affected members (encrypted with the old $TEK_{i,v}$) to the existing members and unicast the new $TEK_{i,v}$ to the new joining member $M_i$ from $SG_i$ encrypted by $M_i$ derived private key $S_{Mi,AKDv}^{KEY}$. This incurs the two rekeying messages at the target cluster $SG_v$ to ensure backward secrecy when $M_i$ joins, i.e.

$$E[J_{dv}] = 2 \quad \ldots\ldots 8)$$

However for $n$ members moving while participating in $m$ $SG_s$, the rekeying overhead at $SG_v$ gives

$$E[J_{dv}] = O(n) \quad \ldots\ldots 9)$$

When member(s) joins $SG_v$ on handoff, the target $LGKS_v$ distribute the new $TEK$ to all the affected members through multicast message encrypted with the old $TEK$. Hence,

$$E\left[IJ_{pv}\right] = n \quad \ldots\ldots\ldots\ldots 10)$$

Therefore, the old $TEK$ of affected $SG_i$ becomes compromised. Therefore, $LGKS_i$ distributes the new $TEK$ to members which is encrypted under individual private key of the $\frac{\lambda_i}{\mu_i}(d_i)$ member's i.e.

$$E[IL_{di}] = d_i \ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots 11)$$

Therefore, the total overhead induced by merging SG*i* and SG*v* on member handoff can be evaluated as

$$TO_{i,v}^{(m)} = RO_T + E[M_{i,v}] \ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots 12)$$

The total overhead $TO_{i,v}^{(m)}$ is estimated as the expected number of rekey messages per unit time due to join or leave in cluster $SG_i$ and $SG_v$ respectively.. However, due to the independent keys for each cluster adopted in DM-GKM, member only considered backward secrecy because members need to maintain the group session continuously at move, therefore, making the forward secrecy becomes jobless in old SG*i* i.e. $E[L_{pi}] = 0 \; and \; E[IL_{pi}] = 0.$

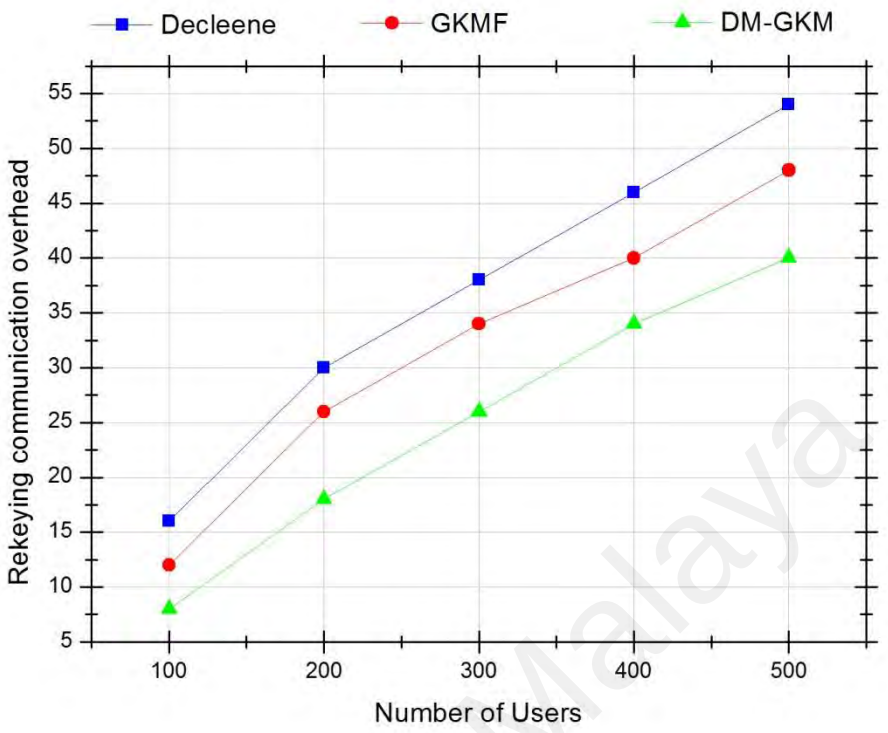


**Figure 5.3:** Communication overhead

Figure 5.3 shows the communication overhead of different schemes and their comparison with DM-GKM. The comparison shows that the communication overhead is significantly reduced when compared to other conventional GKM protocols such as GKMF (Kiah & Martin, 2007), Decleene et al. (DeCleene et al., 2001).

The evaluation of the rekeying message overhead is considered as the leave operation from SG*i* and the rekeying messages overhead induced for join operation in SG*v*. The handoffs member still maintains the group session continuity. Table 5.2 summarizes the rekey overheads induced in DM-GKM. It can be observed from table 5.2 that the convectional schemes induce high $E[M_{i,v}]$ due to *1-affect-n* phenomenon. Therefore, when the TEK changes in any SG, the entire SG needs a new TEK and its affect by rekeying process.

**Table 5.2:** Comparison of communication overhead with other GKM frameworks

| **Reference frameworks** | | **Pairwise approaches comparisons** | | | **LKH approaches** |
|---|---|---|---|---|---|
| | | **Communication overhead at area level** | | **Total Communication Overhead** | |
| | | **$SG_i$** | **$SG_v$** | | |
| Decleene (C. Zhang et al., 2002) | BR | $2 \times [O(n) + 1 + 1]$ | $[O(n) + 1 + 1]$ | $2O(n) + 5$ | $2Olog_d(n) + 5$ |
| | IR | $2 \times [O(n)]$ | $[O(n) + 1]$ | $2O(n)$ | $2Olog_d(n)$ |
| | FEDRP | 0 | $[O(n) + 1]$ | $n\,O(n)$ | $n(Olog_d(n))$ |
| GKMF (Kiah & Martin, 2007) | | 0 | $[O(n) + 1]$ | $O(n)$ | $(Olog_d(n))$ |
| KELLIL (Kellil et al., 2004) | | 0 | $[O(n) + 1]$ | $O(n)$ | $Olog_d(n)$ |
| DM-GKM | | 0 | $O(n)$ | $O(n)$ | $Olog_d(n)$ |

However, DM-GKM, $E[M_{i,v}]$ is performed well by performing rekeying only at the target SG. Moreover, it can also be observed from Table 5.2 that the LKH based rekeying tree based approaches provide the best to reduce rekeying communication overhead from $O(n)$ to $O(log_d(n))$. Hence, DM-GKM introduces lightweight and efficient communication overhead for dynamic mobile members.

### 5.1.3 Bandwidth utilization

The bandwidth overhead measured due to the new keys generated and transmitted, which takes most of the computational power and rekeying overhead. The SG level

consists of limited bandwidth devices and subject to high packet loss, therefore it is important to reduce the rekeying overhead on SG level, so that there is less bandwidth consumption on resource limited devices. By allowing the DGKS to computes the rekeying process gives the LGKS and mobile member's scalability and reduces the signaling overhead at the SG level and on the resource limited mobile devices. Hence, this scenario reduced the rekeying signaling pressure on the intermediate nodes. The DM-GKM is considered as the signaling optimized for efficient bandwidth utilization. The analysis of Markov model can be extended to estimate the bandwidths utilization as demonstrated by (Chan & Chan, 2002). Assuming there is number of handover members M*i* in the systems, which are equally likely to belong to *'m'* number of SGs. Thus the average number of concurrent members per SG, the *m* can be expressed as $\frac{p}{m} = \lambda/\mu m$. Given the expected number of rekey messages per second, then (5) can be re-written as:

$$E\left[R_{\frac{\lambda}{\mu m}}\right] = \lambda/m(E\left[L_{\frac{di}{m}}\right] + E\left[J_{\frac{dv}{m}}\right]) \quad \text{13)}$$

Assuming $B_T$ and $B_S$ define the system total bandwidth utilization in bits/s. Let $R_C$ define a constant rekey message size in bits (which is the sum of the rekey messages data and multicast data). R bits/s is the download data rate for a stream. The total system bandwidth $B_T$ consumption can be achieved by

$$B_T = mB_S + mR_C E\left[R_{\frac{\lambda}{\mu m}}\right] \quad \text{14)}$$

The value of $B_T$ can be optimized by adjusting *m*. By taking in to account the membership dwell time, and multiplying (14) by 1/u time units such that

$$\frac{E\left[R_{\frac{\lambda}{\mu m}}\right]}{\mu} = \mathrm{d/m}(\mathrm{E}\left[L_{\frac{\lambda i}{m\mu i}}\right] + \mathrm{E}\left[J_{\frac{\lambda v}{m\mu v}}\right]) \ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots 15)$$

Therefore $B_T$ in equation (14) can be rewritten as

$$B_T = mB_S + m\mu R_T f(m) \ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots 16)$$

Now dividing (16) by $B_S$ equally gives

$$\frac{B_T}{B_S} = m + m\beta F(m) \ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots 17)$$

Where $\beta = \frac{uR_T}{B_S}$ is the dimensionless parameter. Hence, from (17) it can be deduced that $B_T$ can be obtained by knowing the values of $\mathrm{E}\left[L_{\frac{\mathrm{di}}{\mathrm{m}}}\right] + \mathrm{E}\left[J_{\frac{\mathrm{dv}}{\mathrm{m}}}\right]$. However in DM-GKM, the rekeying of the switching member considers only backward secrecy hence making $\mathrm{E}\left[L_{\frac{\mathrm{di}}{\mathrm{m}}}\right] = 0$. This optimizes $B_T$ which is given as $\mathrm{E}\left[J_{\frac{\mathrm{dv}}{\mathrm{m}}}\right]$ at the target $\mathrm{SG}_v$.

It is important to know the bandwidth of different entities in DM-GKM. The bandwidth is the main source to deliver the rekeying communication (which is the sum of rekey messages plus signaling load) between different entities in the system. Figure 5.4 shows the bandwidth utilization for DM-GKM framework for each entity i.e. member, LGKS and DGKS. The results show that DM-GKM outperforms the conventional protocols by giving the rekeying transmission for increasing number of handoffs. Therefore proposed framework can be considered as signaling optimizer for efficient bandwidth utilization.

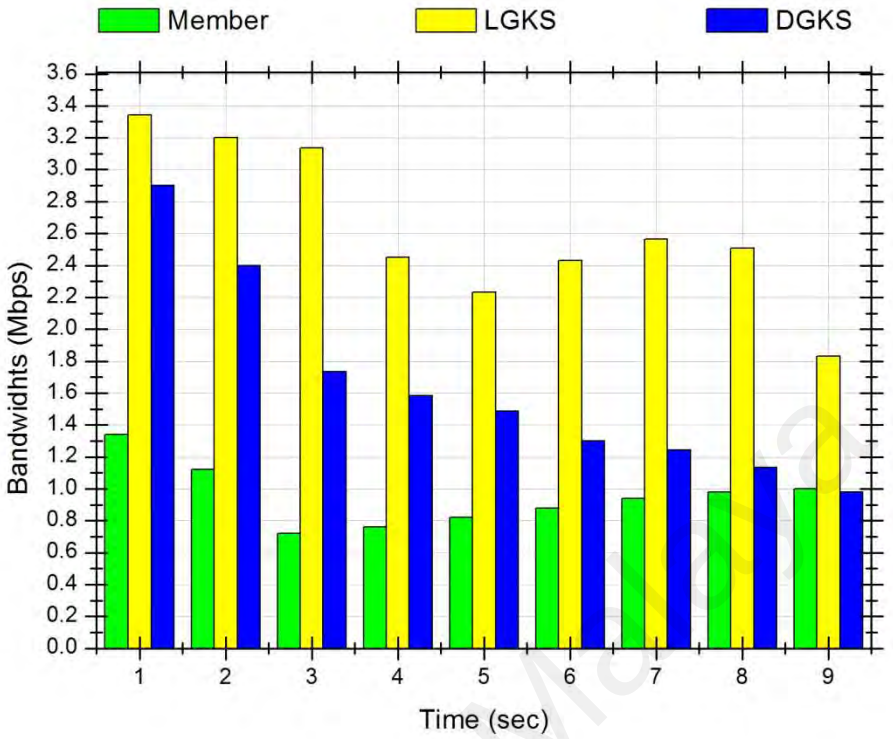


**Figure 5.4:** Bandwidth utilization of different network entities

### 5.1.4 Storage Overhead

Storage overhead is determined by calculating the memory capacity required to maintain the keys by each entity i.e. *DGKS, LGKS* and members of the groups. It is directly propositional to the number of keys if the key sizes are the same (Y. Sun & K. Liu, 2007). This requirement enables the fast execution of the processing and accessibility of the stored keys if there are fewer keys held by each network entities.

Let's suppose the keys arrangement in LKH based approaches (Wallner et al., 1999) (Kwak et al., 2006; Ng & Sun, 2005; Pegueroles & Rico-Novella, 2003) in which each group member required a group key on each level. Therefore, the number of group keys that a group member should have to store is equal to the height of the key tree structure. In case of a binary tree, $n = 2^H$, the height of a binary tree is $H$. Thus, the group member has to store $H$ number of group keys. Similarly, for a full hierarchical structure

with $H$ levels of hierarchy, the group controller stores $\left(\frac{d^h-1}{d-1}\right)(2^d-1)$ keys and each user stores $h.(2^{d-1})$ keys. Thus, in the hierarchical structure, for small values of d, the user needs to store $O(h)$ keys (Amir et al., 2004). Figure 5.5 demonstrates the storage overhead of DM-GKM with respect to the number of members in the system.

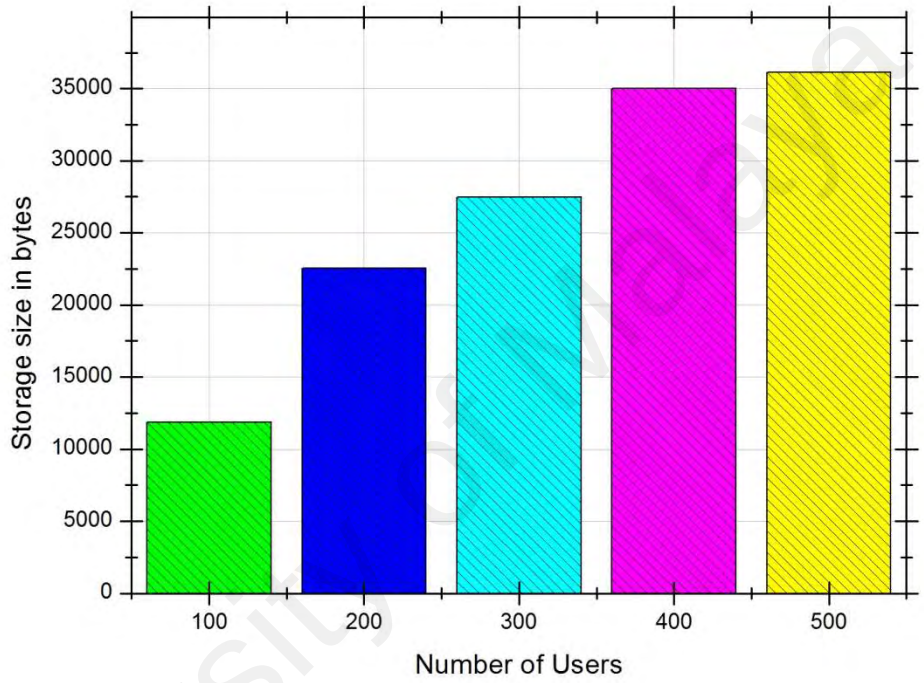


**Figure 5.5:** Storage size of main server vs no of users

The DM-GKM structure is well organized that it has the enough memory space to store enough number of group keys and user's private keys. Even though a group member's device has very small storage, the DM-GKM enables the group member to join the communication group. In proposed scenario, the DGKS should have '*m'* TEKs for '*m*' $SG_s$. Table 5.3 presents the storage overhead of different entities when there is a move from $SG_i$ to $SG_j$. The other GKM schemes add storage complexity to the resource limited mobile devices by introducing extra local area keys for each cluster. However, the DM-GKM adopts independent TEK per SG without extra key memory required.

**Table 5.3:** Comparison of storage cost

| Storage cost | DeCleene (DeCleene et al., 2001) | GKMF (Kiah & Martin, 2007) | KELLIL (Kellil et al., 2004) | DM-GKM |
|---|---|---|---|---|
| At DGKS | n | n+2 | n | n |
| At LGKS | n+1 | n+4 | n+2 | n+1 |
| At member | n+1 | n+3 | n+3 | 3 |

In DM-GKM, the handoffs users only revoke the cryptography keys of the previously visited cluster upon complete handoffs, hence giving the scalability to DGKS and LGKS. Therefore, members visiting in multiple clusters do not need to triggers the rekeying process. Such as in the approaches of DeCleene et al (DeCleene et al., 2001), GKMF (Kiah & Martin, 2007) and Kellil et al (Kellil et al., 2004) triggers the rekeying in all the clusters. Thus, introduces high storage overhead at each entity due to calculating the more encryption keys which can takes more battery power for the resource limited mobile devices. This implies that, the users may loss connections with servers when moving in multiple SGs in dynamic wireless mobile environment. Hence, members do not need extra storage requirement as the members' only need to maintain the keys of that particular cluster in which they are currently resides. As a result, there is very less storage and communication with the involving communicating parties. Figure 5.6 provides the comparison of the number of keys required by each GKM schemes.

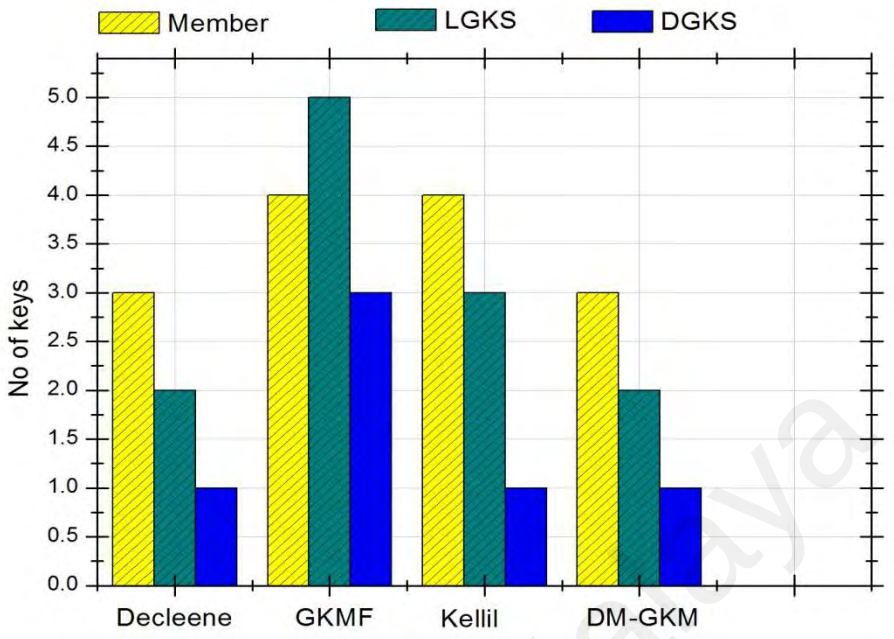


**Figure 5.6:** Storage Overhead

The switching members only need to maintains three keys regardless of locations which makes DM-GKM adaptive to dynamic group based applications as compared to other schemes GKMF (Kiah & Martin, 2007), DeCleene (DeCleene et al., 2001)and m-Iolus (Mittra, 1997). Figure 5.7 shows the comparison of number of keys maintained by each network entity when there are '*n'* number of handoff members in the group (where *n*=10). From analytical model and simulation results it can be concluded that DM-GKM incurs less storage overhead at the member, LGKS and DGKS as compared with other convectional GKM schemes.

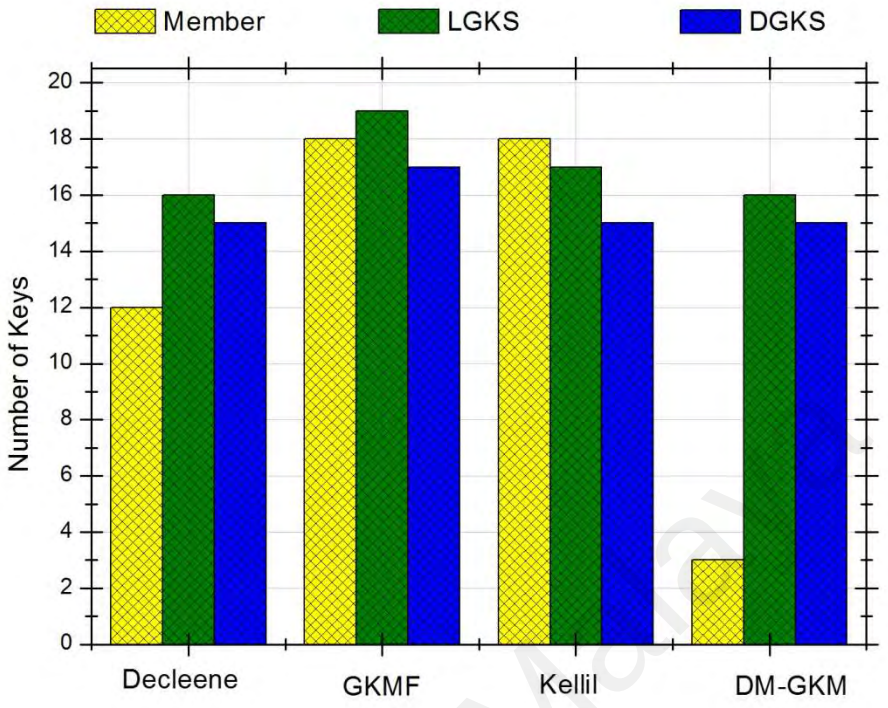


**Figure 5.7:** Storage of no of keys upon member's handoffs

### 5.1.5 Computational rekeying overhead

This Section evaluates the computational cost of the proposed DM-GKM framework. However, the major problem still exists which is the computational complexity of calculating the encryption and decryption of the proposed framework. Since DM-GKM uses asymmetric encryption, which needs more computational complexity as compared to symmetric methods, therefore it is important to check the feasibility of the proposed DM-GKM in real environment it terms of computation power.

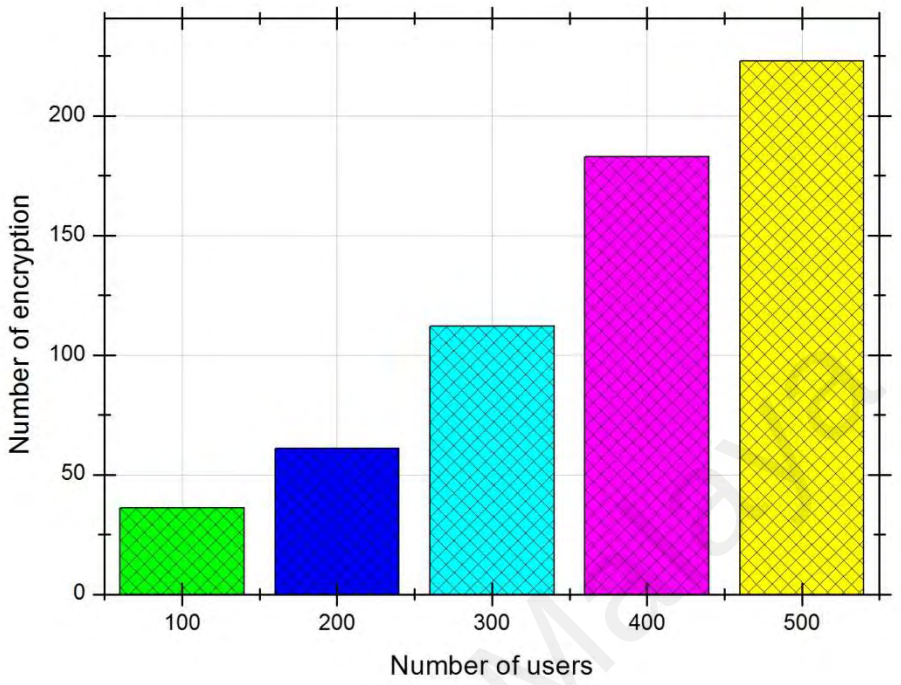


**Figure 5.8:** Number of encryptions vs number of users

The computational overheard occurs when there is key generation from the DGKS for the entire system. Since the key generation occurs only one time during preparation step of the group setup, therefore, it does not have any effect on the real time rekeying process. Moreover, when the DGKS updates the keys due to membership changes, this modification does not affect the whole system, because the modification occurs in only local SGs where membership change occurs. Therefore, it is only need to modify the keys at the complexity of $O(n)$, hence it may not be overburden on the entire system. Figure 5.8 demonstrated the no of encryption performed by the DM-GKM with respect to the number of users.

Key management technique which is uses in this framework is the number of encryption and decryption of messages. This implies that, the decryption is done for the received messages using the per-group TEK and KEK of its encryption area. After this

message is send to the members of its area. In DM-GKM, the number of key generations at join operation is 2 and at leave is 1. It is because at join, the new member private key and group key is generated, and at leave, only new group key need to be generated by the DGKS. DM-GKM reduces key generation and key encryption overhead largely at join/leave operation compared to other GKM schemes.

In DM-GKM, the encryption/ decryption only happens when there is membership changes in each subgroup, as compared to symmetric keys encryption where each encryption/ decryption happens on each delivery of the user data. Such as GKMF (Kiah & Martin, 2007), Decleene et al (DeCleene et al., 2001). and Kellil (DeCleene et al., 2001) gives invokes on every membership changes, therefore it increases the computational overhead at area and domain level. As a result the storage overhead is also increases at area level at the expense of computational load. Therefore, DM-GKM can be considered as efficient signaling for optimized the bandwidth utilization. Hence, we can conclude that the DM-GKM framework do not suffer from additional computational overhead.

Figure 5.9 shows the number of affected members when there is some event happens i.e. group member change. The figure indicates that when the number of users' increases there is more members affected in other approaches. However, DM-GKM incurs less rekey messages and computational overhead due to the smart arrangement of users and keys, hence, giving scalability to the entire system.

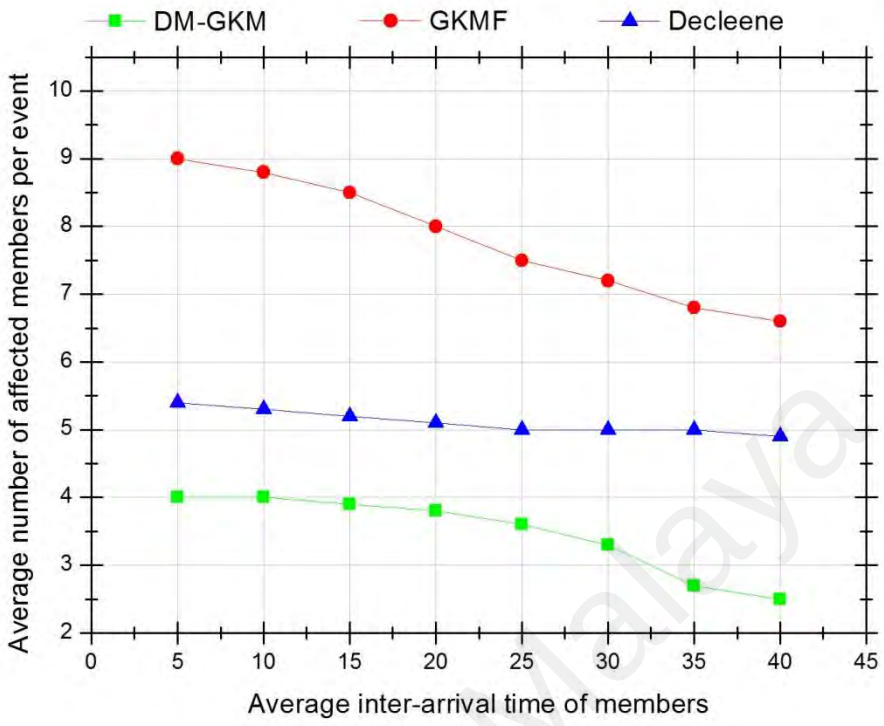


**Figure 5.9:** Average number of affected members per event

DM-GKM framework handles unique assumptions that the resource constrained mobile devices should be less affected by any event occurs inside the group. As the CRT method is combined with RSA for decryption process, therefore, it takes less computation overhead on mobile devices by decrypting the messages. Therefore, it can save the power of battery of resource constrain mobile devices which other schemes don't. The simulation results in Figure 5.10 demonstrate that the DM-GKM takes very less computational power for decrypting the messages. Hence, giving scalability to both LGKS and DGKS.

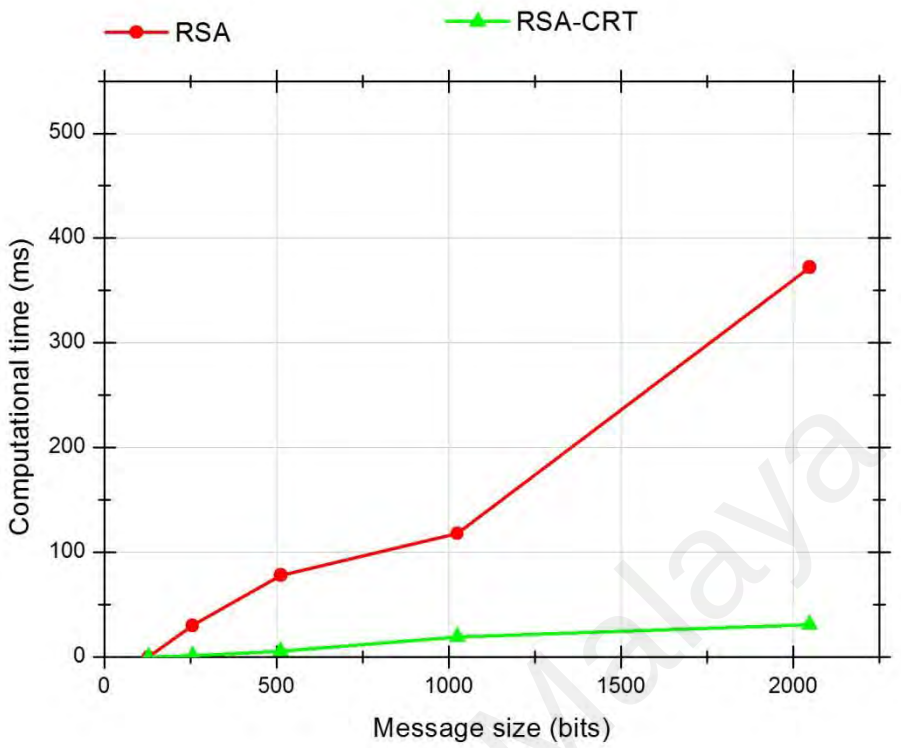


**Figure 5.10:** Decryption time of RSA vs RSA-CRT methods

### *a) Real-time key computation scenario*

Let's estimate the average computation time of new key generation in DM-GKM. In simulation experimental results, the average computation time is .004 sec for new key generation. In other words, the DM-GKM can support the membership change rate of at least 238 (≈ 1/.004) per second, which is equivalent to at least 20,563,200 per day. If there are 25% members changes their membership in a day, the DM-GKM framework can accommodate 82,252,800 users, which demonstrates the scalability of the DM-GKM scheme. In addition, the rekeying performance can still be increases by using the high-end powerful system. Therefore, it can be concluded that the computational overhead of the DM-GKM is acceptable in the sense that it can support real-time key computation update for a reasonable number of wireless mobile users.

## 5.2 Statistical Analysis

To verify the results, the Statistical Package of Social Sciences (SPSS) 24.0 (IBM, 2017) for data analysis is used in this study. The normality test is conducted in order to show the equally distribution of data, whereas, the regression analysis shows the relationship among variables.

### 5.2.1 Normality test

The Normality test is performed to see whether the data is equally distributed. The null hypothesis is that the data is normally distributed and the alternative hypothesis is that the data is not normally distributed. The results from the two continuous variables "storage overhead" and "no of encryption" are taken to check the normality of this data set. Table 5.4 presents the descriptive analysis of the experimented data.

**Table 5.4:** Descriptive analysis of the data

| **Descriptive** | | | | |
|---|---|---|---|---|
| | | | Statistic | Std. Error |
| No_of_Encryption | Mean | | 132.63 | 19.224 |
| | 95% Confidence Interval for Mean | Lower Bound | 93.32 | |
| | | Upper Bound | 171.95 | |
| | 5% Trimmed Mean | | 126.57 | |
| | Median | | 115.00 | |
| | Variance | | 1.109E4 | |
| | Std. Deviation | | 105.295 | |
| | Minimum | | 3 | |
| | Maximum | | 390 | |
| | Range | | 387 | |
| | Interquartile Range | | 140 | |
| | Skewness | | .732 | .427 |
| | Kurtosis | | -.091 | .833 |
| Storage_Overhead | Mean | | 2.1300E4 | 2.52976E3 |

| Descriptive | | | | |
|---|---|---|---|---|
| | 95% Confidence Interval for Mean | Lower Bound | 1.6126E4 | |
| | | Upper Bound | 2.6474E4 | |
| | 5% Trimmed Mean | | 2.1416E4 | |
| | Median | | 2.3409E4 | |
| | Variance | | 1.920E8 | |
| | Std. Deviation | | 1.38561E4 | |
| | Minimum | | 867.00 | |
| | Maximum | | 3.99E4 | |
| | Range | | 3.91E4 | |
| | Interquartile Range | | 2.82E4 | |
| | Skewness | | -.194 | .427 |
| | Kurtosis | | -1.452 | .833 |

For the case of "No_of_Encryption" the value of mean is 132.63 and the value of median is 115, these values are close to each other. Similarly for the case of "Storage_Overhead" the value of mean is 2.1300 and the value of median is 2.340 that are close enough to each other.

**Table 5.5:** Normality test

| Tests of Normality | | | | | | |
|---|---|---|---|---|---|---|
| | Kolmogorov-Smirnov[a] | | | Shapiro-Wilk | | |
| | Statistic | df | Sig. | Statistic | df | Sig. |
| No_of_Encryption | .116 | 30 | .200* | .930 | 30 | .048 |
| Storage_Overhead | .148 | 30 | .090 | .896 | 30 | .007 |
| a. Lilliefors Significance Correction | | | | | | |
| *. This is a lower bound of the true significance. | | | | | | |

The results mentioned in Table 5.5 compares the scores in the sample to a normally distributed set of scores with the same mean and standard deviation; the null hypothesis is that "sample distribution is normal." We can see from the Table 5.5 that both the p-value greater than 0.05, which indicates normal distribution of data.

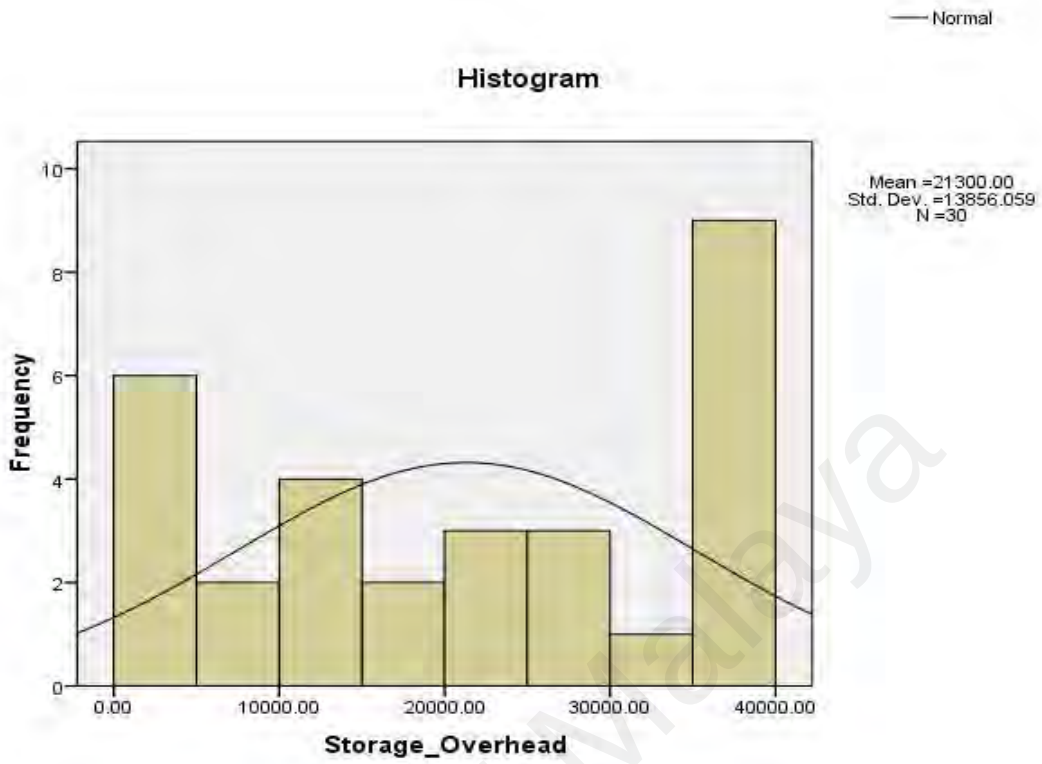


**Figure 5.11:** Histogram of storage overhead

The Figure 5.11 shows that the common pattern is the bell–shaped curve known as the "normal distribution". It is demonstrated that there is no significance outliers and the sample is of adequate size, therefore, there is linear relationship between the variables. Moreover, from Figure 5.12 it is also demonstrated the Q-Q plot; which shows that the

data plausibly came from some theoretical distribution such as a Normal distribution.

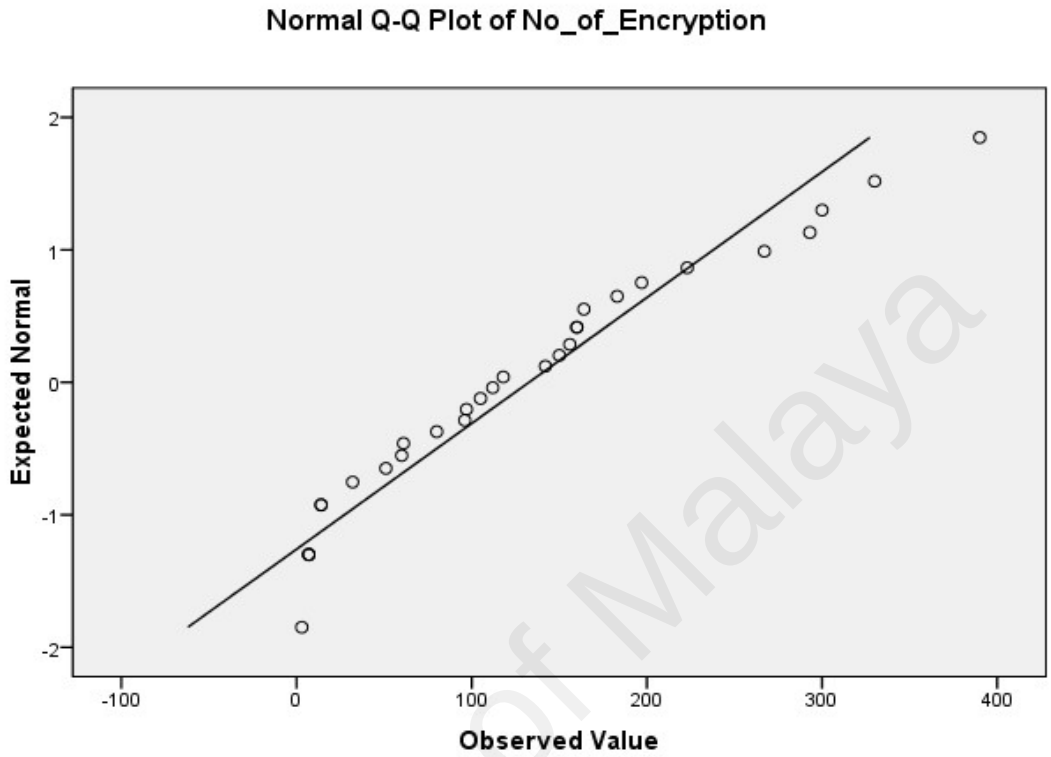


**Figure 5.12:** No of encryption

From the Figure 5.12 and Figure 5.13 it can be seen that the variables are normally distributed, therefore the parametric test can be applied.

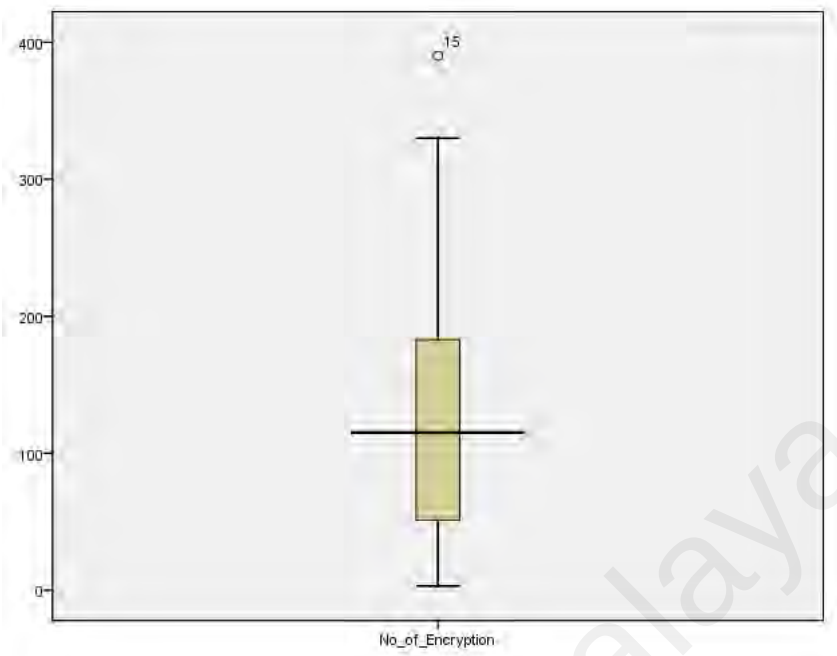


**Figure 5.13:** Analysis of number of encryption

Results from Table 5.6 shows that the value of $p<0.001$ and $r=.825$. These results demonstrated that there is significance and almost good correlation between no of encryption and storage overhead in proposed DM-GKM. Hence, from the detail statistical analysis it can be concluded that the data is equally distributed.

**Table 5.6:** Correlations between attributes

| Correlations | | | |
|---|---|---|---|
| | | No_of_Encryption | Storage_Overhead |
| No_of_Encryption | Pearson Correlation | 1 | .825** |
| | Sig. (2-tailed) | | .000 |
| | N | 30 | 30 |
| Storage_Overhead | Pearson Correlation | .825** | 1 |
| | Sig. (2-tailed) | .000 | |
| | N | 30 | 30 |
| **. Correlation is significant at the 0.01 level (2-tailed). | | | |

### 5.2.2 Regression Analysis

Regression analysis is a statistical process for estimating the relationships among variables. Table 5.7 demonstrated that the multiple correlation coefficients are 0.921892556, which is equal to 1. This indicates that the correlation among the independent and dependent variables is positive.

**Table 5.7:** Regression Analysis

| *Regression Statistics* | |
|---|---|
| Multiple R | 0.921892556 |
| R Square | 0.849885885 |
| Adjusted R Square | 0.737300299 |
| Standard Error | 0.24216081 |
| Observations | 15 |

The coefficient of determination, R2, is 84.9885%. This means that it is close to 85% of the variation in the dependent variable. Hence it can be explained by the independent variables. Moreover, Figure 5.14 shows the predictive vs linear scenario for bandwidths and number of users.

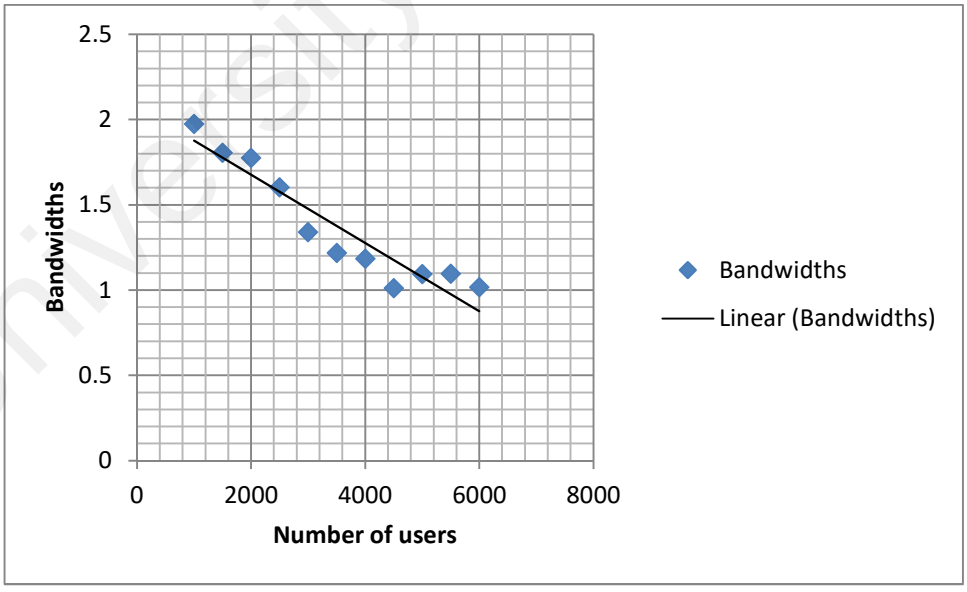


**Figure 5.14:** Scatter Analysis of Bandwidths and number of users

## 5.3 Security Analysis

This Section analyses the security requirement which are necessary for secure group based applications. These security requirements based on confidentiality, integrity, anonymity, non-repudiation, replay attacks prevention, and man-in-the-middle attacks prevention.

### 5.3.1 Confidentiality

When there is communication messages exchange between entities in secure group session, these are encrypted with asymmetric keys. Therefore the messages remain the confidential. The shared key between the main server and member is also encrypted. In this situation, the attackers are unable to access the encrypted messages, therefore, they are unable to acquire the information exchange between member and main server. Hence the information exchange is only accessible to authorize entities only. Suppose in the situation when the adversaries can eavesdrop all traffic. However, they don't have any key because they don't belong to any of the group. So, the adversaries are unable to obtain user's private key from the ciphertext without GK. Moreover, because the adversaries are unware of previous GK and the rekeying material, they cannot deduce public key and private keys by themselves. Therefore, the data confidentiality is guaranteed.

### 5.3.2 Anonymity

During the process of messaging with server and member, mutual authentication is only performed with the key server. Therefore only the key server knows the members real identity. In this way the server issues private IDs to the member. By using this ID the member can performs his task securely.

### 5.3.3 Backward security

When the new user joins the group, all the keys including the group key of that group are updated. Because the new keys are only shares with the group members the newly joining member cannot be able to compute the previous keys from the new keys. In this way, backward security is guaranteed.

### 5.3.4 Forward security

When a user leaves the group, the core network generates a new GK and transmits it to the remaining members of the group via a secure channel. Therefore, the departed user cannot obtain the new GK. Subsequently, the new keys are broadcast to the remaining users after being encrypted with new GK. Although, the malicious members can obtain the secret keys of the member, they still unable to get new GK of the rekeying material. Because any private key and the corresponding randomly selected secret keys of the members are unknown to the malicious users. It is difficult to obtain the valid keys of the group and members.

### 5.3.5 Non-repudiation

All the messages are encrypted in our framework and are signed. When a user joins the group it's all the keys are generated for it and identification. The pair of keys i.e. the public and the privates' keys are generated for it. Therefore, the procedure of key generation guarantees the non-repudiation.

### 5.3.6 Replay attacks prevention

The proposed framework encrypted every message with nonce. With the change on random values in each session, hence the attackers cannot replay previous messages to

break the authentication procedure. In fact, it is very difficult to the attackers to replay the messages that the server uses for the key generation process because all the messages are encrypted. Asynchronous replay attacks are also prevented due to the nonce. After the member leaves, the proposed framework ensures that these old keys cannot be used for other members. Therefore the attackers also cannot replay another member's identity.

### 5.3.7 Integrity

When a user joins the group, the server generates the public-private keys pairs for the group. Because the key pair is unique and change in every session, attackers cannot modify it. Hence the integrity of the message is ensured due to the public keys and user private keys pairs. Only the validate members with their private keys are able to decrypt the messages. All the communications are encrypted to prevent forgery and modification.

## 5.4 Security Analysis using BAN logic

The BAN logic was named after its inventors, Mike Burrows, Martiın Abadi, and Roger Needham (Kyntaja, 1995). The stated it as the logic of belief and action. These methods are the set of rules for defining and analyzing information exchange protocols (Burrows, Abadi, & Needham, 1989).

The BAN logic is used here to prove the non-repudiation, message integrity, and authentication. When a receiver gets a message, he/she can construct the session key and decrypt the ciphertext to confirm whether the nonce is valid. Therefore, we assume that all participants in this mechanism can verify the correctness of the nonce and believe in the freshness of nonce N. Nonce stands for number which only used once. It a

random number issued in an authentication protocol to ensure that old communications cannot be reused in replay attacks. The value of nonce helps the users to be sure that the message is recently sent. In this scenario, *K* is a key which is generated by two participants. *A* and *B* need to communicate together by secured *Kab* by the server's recommendation. *A* and *B* trust the server *S* which produces the key during the execution of the protocol. Communications between the server *S* and participants *A* and *B* will be by *Kas* and *Kbs* respectively; the keys have to be known by only both participants.

### 5.4.1 Syntax and Semantics of BAN Logic

*A, B* are the interaction principal, *S* is a authentication server; $K_{ab}$, $K_{as}$, $K_{bs}$: shared keys between principals; $K_a$, $K_b$: public keys of principals; $K^{-1}{}_a$, $K^{-1}{}_b$: secret keys of principals; $N_a$, $N_b$: formulas or statements of principals; P, Q: principal variables; X, Y: statements in general; K: encryption keys; (X, Y): conjunctions of X and Y.

P |≡ X: P believes X, the principal P trusts the message X is true.

P ▷X: P sees X, P received a message containing X and P can read and repeat X.

P | ~X: P said X. The principal P at some time sent a message including the statement X. There are two intendments: one is that the message X is sent by P, that is the message source is P; the other is that the principal P can confirm and discern the message X and explain the message X correctly.

P |⇒X: P has jurisdiction over X.

#(X): The formula X is fresh; that is, X has not been sent in a message at any time before the current run of the protocol.

Q: P and Q may use the shared key K to communicate each other. The key K is good, that is, the key is exclusive and will never be discovered by any principal except P or Q, or a third principal trusted by either P or Q.

(K)→P: The key K is the public key of principal P;

P ⇌ (x) Q: The formula X is a secret known only to P and Q, and any other principals do not know the X except P and Q and the principals trusted either by P or Q.

{X}K: This represents the formula X encrypted under the key K.

<X> Y: This represents X combined with the formula Y; it is intended that Y be a secret.

### 5.4.2 Basic notations and assumptions of BAN logic

Tables 5.8 briefly present the basic notations and semantics of BAN logic model.

**Table 5.8:** BAN logic symbols and notations

| Notations | | Definitions |
|---|---|---|
| (N1) | X | Statement |
| (N2) | P,Q | Participants |
| (N3) | P \|≡ X | P believes in X |
| (N4) | P◃X | P sees X |
| (N5) | P\|~X | P once said X |
| (N6) | P\|⇒X | P has the jurisdiction over X |
| (N7) | #(X) | Formula X is fresh |
| (N8) | (K)→P | The key K is the public key of principal P |
| (N9) | P↔(k)K Q | P and Q may use the shared key K to |

| | | |
|---|---|---|
| | | communicate |
| (N10) | P⇌(Y) Q | Formula Y is a secret known to P and Q |
| (N11) | $\{X\}_K$ | Formula X encrypted under the key K |
| (N12) | <X>Y | Statement X combined with formula Y |
| (N13) | P/Q | If P is true then Q is also true. |

### 5.4.3 Rules of BAN logic

1. Message Meaning rule

For shared key:

$$\frac{P| \equiv Q \overset{K}{\leftrightarrow} P, P\ \Delta\ \{X\}_K}{P| \equiv Q\ | \sim X}$$

For public key:

$$\frac{P| \equiv \overset{K}{\rightarrow} Q, P\ \Delta\ \{X\}_{K^{-1}}}{P| \equiv Q\ | \sim X}$$

For shared secret message:

$$\frac{P| \equiv P \overset{K}{\leftrightarrow} Q, P\ \Delta\ \{X\}_Y}{P| \equiv Q\ | \sim X}$$

Nonce Verification rule:

$$\frac{P| \equiv \#\ (X), P| \equiv Q,\ | \sim X}{P| \equiv Q\ | \equiv X}$$

2. Seeing rules:

$$\frac{P\Delta\,(X,Y)}{P\Delta X}$$

3. Rule:

$$\frac{P\Delta < X >_Y}{P\Delta X}$$

4. Rule:

$$\frac{P| \equiv \mathrm{P} \overset{K}{\leftrightarrow} Q, P\ \Delta\ \{X\}_K}{P\ \Delta\ X}$$

5. Rule:

$$\frac{P| \equiv \overset{K}{\rightarrow} Q, P\ \Delta\ \{X\}_K}{P\ \Delta\ X}$$

6. Rule:

$$\frac{P| \equiv \overset{K}{\rightarrow} Q, P\ \Delta\ \{X\}_{K^{-1}}}{P\ \Delta\ X}$$

7. Freshness rules:

$$\frac{P| \equiv \#\,(X)}{P| \equiv \#(X,Y)}$$

8. Belief rule:

$$\frac{P| \equiv X, P\ | \equiv Y}{P| \equiv (X,Y)}$$

9. Jurisdiction rule:

$$\frac{P| \equiv Q \Rightarrow X, P \mid \equiv Q| \equiv X}{P| \equiv X}$$

10. Session key rule:

$$\frac{P| \equiv \#(X), Q \Rightarrow P \mid \equiv Q| \equiv X}{P| \equiv P \overset{K}{\leftrightarrow} Q}$$

### 5.4.4 BAN logic assumptions

The following assumptions can be made for the proposed scenario.

**A1:** Ui |≡ (SGi, SGj)

**A2:** S |≡ (SGi, SGj)

**A3:** Ui |≡ Ui $\overset{MK}{\leftrightarrow}$ S

**A4:** S |≡ Ui $\overset{MK}{\leftrightarrow}$ S

**A5:** Ui |≡ # (PKi, PKj)

**A6:** Ui |≡ Ui $\overset{K}{\leftrightarrow}$ S

**A7:** U|≡ $\overset{K_C}{\rightarrow}$U

**A8:** U|≡ $\overset{K_S}{\rightarrow}$S

**A9:** S|≡ –K (u) →S

**A10:** U|≡ S⇒U

### 5.4.5 Goals of BAN logic

There are number of goals which should be achieved for the proposed protocol scenario.

Goal 1: Ui |≡ (Ui $\overset{K^-}{\leftrightarrow}$ S)

Goal 2: Ui |≡ S |≡ (Ui $\overset{K^-}{\leftrightarrow}$ S)

Goal 3: A |≡ A $\overset{K^-}{\leftrightarrow}$ B

Goal 4: A |≡ A $\overset{K^+}{\leftrightarrow}$ B

Goal 5: A |≡ # (K)

Goal 6: A |≡ (B |≡ B $\overset{K^{-1}}{\longleftrightarrow}$ A)

Goal 7: A |≡ A $\overset{SK}{\leftrightarrow}$ A

Goal 8: B |≡ A $\overset{SK}{\leftrightarrow}$ A

Goal 9: A |≡ B |≡ A $\overset{SK}{\leftrightarrow}$ B

Goal 10: B |≡ A |≡ A $\overset{SK}{\leftrightarrow}$ B

### 5.4.6 Idealized form of the protocol

I1: Ui → S

I2: S → Ui

I3: SG → S:

I4: S → SG:

By using the assumptions, goals and rules, the idealized form, the security of the proposed framework is analyzed using the BAN logic rules. The main procedure of the proof as follows:

From (I1) we get the following:

S1: $A|\equiv (PK_A, SK_A)$

S2: $B^{\triangleleft} (PK_A, SK_A)$

S3: $B^{\triangleleft} \{PK_A, SK_A\}d_A$

From (I2) we get the following:

S4: $B|\equiv (PK_B, SK_B)$

S5: $A^{\triangleleft} (PK_B, SK_B)$

S6: $A^{\triangleleft} \{PK_B, SK_B\}d_B$

From S2: $B^{\triangleleft} (PK_A, SK_A)$ and A2 $B|\equiv \xrightarrow{P_A} A$

S7: B |≡ A|~(PKA, SKA)

From assumptions $A|\equiv \#(PK_A)$ and $A|\equiv \#(S_A)$

S8: A |≡ # (PKA, SKA)

From S7 and S8, we get

S9: B |≡ A |~ ($PK_A$, $SK_A$)

From S9 and using the synthetic rule, we get

S10: B |≡ ($PK_A$, $SK_A$)

From S7, S10 and through the nonce verification rule, we get

S11: B |≡ A |≡ ($PK_A$, $SK_A$)

On applying the belief rule, we get

S12: B |≡ A |≡ ($PK_A$)

S13: B |≡ A |≡ $SK_A$

Applying the jurisdiction rule and from equation S12 and assumption B |≡ A ⇒ ($PK_A$)

S14: B |≡ ($PK_A$)

Applying the jurisdiction rule and assumptions B |≡ A ⇒ ($SK_A$) and S13, we get

S15: B |≡ ($SK_A$)

From S10 and applying the freshness rule,

S16: B |≡ # ($SK_B$)

Also from S16

S17: B |≡ # ($SK_A$)

From S6 and assumption A $|\equiv \xrightarrow{P_A}$ B, applying the message meaning rule, we get:

S20: A $|\equiv$ B $|\sim$ ($PK_B$, $SK_B$)

From assumptions B $|\equiv \xrightarrow{P_B}$ B and B $|\equiv$ # $R_B$ and through freshness rule, we get:

S21: B $|\equiv$ # ($PK_B$, $SK_B$)

Thus, we have successfully proved the goals (Goal 1 to Goal 10) of the proposed framework corresponding to the equations mentioned above. It can be conclude that the entities involved successfully generate a fresh, common and secure messages and keys between them.

### 5.5 Discussion of results and performance comparison

In this Section, the detail analysis of proposed lightweight key management framework is discussed. The comparison has done by considering different parameters such as storage, communication and computation overhead. We compared these results with existing key management protocols such as GKMF (Kiah & Martin, 2007), Decleene (DeCleene et al., 2001), m-Iolus (Mittra, 1997) and KELLIL (Kellil et al., 2004) model.

As can be seen from the Figure 5.9 and Figure 5.10, M-Iolus and Kellil et al. scheme increases the number of encryptions along with the storage of multiple keys. The protocol also lacks trust relationship due to data transformations which can expose the data to eavesdropping. Moreover, when a member leaves after moving between several SGs, the rekeying trigger and it will affect all the areas, hence adding more control overheads which wastes the bandwidths**.** The DeCleene et al. method suffered from the service disruption in both new and old area during rekeying process as can be seen from

results in Figure 5.11. It is also not able to differentiate between each join and leave of a member from its movement across the areas and suffered from high rekeying overhead, especially in the case if member moves rapidly across the areas. The FEDRP method is best as compared to other in terms of communication overhead, scalable, low communication overhead, and support of highly dynamic membership. However this approach suffers from area key which can be compromised. Moreover, by using the common TEK the key management task can be suffer from 1-affect-n phenomenon. The authentication of the users is also not considered in this approach. Single DKD is used in this approach which can cause the single point of failure. Time synchronization problem can also introduce unnecessary rekeying. Similarly the protocol of Kellil, GKMF and Gharout et al. also suffered from 1-affect-n problem because it uses the common TEK approach. This implies that, these methods cannot be able to handle the highly dynamism and highly mobile members due to multiple rekey overhead. The GKMF approach can suffer from storage overhead for resource constrained mobile devices by using the large number of used keys. The common TEK is used which can cause the 1-affect-n problem. The member moving can suffer from join latencies due to rekey of both area key and common TEK which are done independently. A member which visited the multiple areas can cause key update of both the area key and the traffic key of all the areas. Gharout et al method experience join latency due to data transformation processing delays when inter moves occurs. Therefore, these solutions lack of support for highly dynamic membership handoffs.

The DM-GKM framework can obviously improve the overall system performance by greatly reducing the computation overheads in encryption and decryption operations, provide flexible and expressive data access control policy, and meanwhile enable lightweight key management methods for dynamic mobile users. By offloading the key

management and authentication phases to the core network massively reduced signaling load at the intermediate nodes and resource constrained mobile devices. By doing this arrangement the DM-GKM gives the scalability over conventional schemes while preventing bottleneck. As can be seen by the performance analysis, the DM-GKM focused on the optimization of user computation, number of stored keys, and number of the rekey message with loading certain amount of computation on the DGKS. In contrast with other convectional schemes, DM-GKM used a new rekeying strategy based on lightweight RSA and CRT methods, which effectively performing key management and authentication phases respectively during handoff.

In dynamic group based applications, the security is a challenging issue and is very difficult to handle. BAN logic is used to determine what are the role of each message of a protocol should be and to ensure freshness properties of messages. By analyzing the authentication protocol for DM-GKM with BAN logic, the results demonstrate that the proposed framework can effectively achieve the security goal of mutual authentication of different entities. The performance and formal analysis of DM-GKM proves its suitability in the group based applications systems. Table 5.9 presented the comparison of proposed DM-GKM framework with other approaches by considering different parameters. In this Table '✓' represents the 'yes' and '×' represents 'no'.

**Table 5.9:** Comparison of DM-GKM with other methods

| **Evaluation criteria** | **BR (C. Zhang et al., 2002)** | **IR (C. Zhang et al., 2002)** | **FEDRP (C. Zhang et al., 2002)** | **GKMF (Kiah & Martin, 2007)** | **KELLIL (Kellil et al., 2004)** | **DM-GKM** |
|---|---|---|---|---|---|---|
| Decentralized framework | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| Forward security on member join | ✓ | ✓ | × | ✓ | × | ✓ |
| Backward security on member move | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| Membership changes | × | × | × | × | × | ✓ |
| LGKS to LGKS communication link | × | × | × | × | ✓ | ✓ |
| Single point of failure | ✓ | ✓ | ✓ | ✓ | × | × |

| Evaluation criteria | BR (C. Zhang et al., 2002) | IR (C. Zhang et al., 2002) | FEDRP (C. Zhang et al., 2002) | GKMF (Kiah & Martin, 2007) | KELLIL (Kellil et al., 2004) | DM-GKM |
|---|---|---|---|---|---|---|
| DGKS scalability | × | × | × | × | ✓ | ✓ |
| Use of list to manage mobility | × | × | ✓ | ✓ | ✓ | ✓ |
| Authentication at move | × | × | × | × | ✓ | ✓ |

## 5.6 Chapter summary

This Chapter presents DM-GKM experimental results, performance and security analysis. The comparison is done by considering different parameters such as storage, communication and computation overhead. Moreover, it analyses the performance of DM-GKM and presents a rekey performance comparison with the existing GKM methods. The performance analysis and results comparison shows that DM-GKM has achieved the set objectives and has an edge over the existing GKM protocols in several aspects, namely but not limited to, rekey efficiency, performance, secrecy, storage overhead, and bandwidths optimization. We show that the proposed framework is well efficient and performed well in high mobility scenario.

# CHAPTER 6: CONCLUSION AND FUTURE WORK

This Chapter summarizes the major finding of this research work by reviewing the achievements of the research objectives listed in Chapter 1. The next Section presents the significant of contribution. This is followed by the future work.

## 6.1 Achievement of research objectives

The achievement of research objectives drawn from this study are presented as follows:

a) A comparative study of existing GKMF schemes has been carried out in this research, as presented in Chapter 2 of this study, which fulfils the first research objective. The characteristics of wireless mobile networks, and the architecture and entities that affect the distribution and management of keys are presented. A qualitative comparison of existing GKMPs is also carried out in order to gain a better understanding on the features, advantages and disadvantages of each protocol, with emphasis on host mobility issues in wireless mobile environments. The limitations of the current security solutions used for group based applications are also identified, which form the basis of designing a suitable GKMF for wireless mobile environments. This fulfils the second research objective.

b) A new, improved key management framework has been proposed in this research, which is called the "DynaMic Group Key Management" (DM-GKM) framework. This fulfils the third objective. This framework makes use of the RSA-CRT algorithm and a decentralized architecture. The RSA is a public key cryptography algorithm which accounts for the heavy load of calculations during the encryption process by placing the heavy load at powerful servers whereas the

CRT algorithm accelerates the decryption process at resource-limited mobile devices.

c) The fourth objective is fulfilled by implementing the proposed solution by means of simulation. The DM-GKM framework exploits the benefits of the CRT-RSA algorithm in order to achieve maximum security, confidentiality and authentication. The DM-GKM framework offers great flexibility for group based applications owing to the application of public key cryptography and decentralized architecture. In addition, the DM-GKM framework adapts automatically to account for group dynamism using an independent key for each cluster, which optimizes the signaling load.

d) The performance of the DM-GKM framework has also been evaluated in this study and the results indicate that the framework works well for large, dynamic groups with a significant reduction in the computational, storage and communication overheads. This fulfills the fifths objective

## 6.2 Significance of contribution

The goal of this research is to develop a lightweight GKMF for group based applications in wireless mobile environments. This goal has been attained with the development of the DM-GKM framework, which fulfils the requirements of a lightweight GKMF.

The other contributions of this research are as follows:

- A critical analysis of various GKMPs and cryptographic algorithms has been carried out in this research. The limitations of existing GKMPs are used as the groundwork to design a more robust GKMF (*i.e.* DM-GKM framework) for a large group of users in wireless mobile environments. The DM-GKM

framework exploits the benefits of a public key cryptography algorithm and decentralized architecture and provides superior information security. The DM-GKM framework ensures backward and forward confidentiality when mobile members dynamically perform handoffs while seamlessly maintain the group communication session.

- A reference framework based on a decentralized architecture is also presented in this thesis, which includes an analysis on the performance and security features of the framework. The concept of the DM-GKM framework used for dynamic group based applications is also discussed in this thesis.
- Unlike conventional schemes, the DM-GKM framework offers a lightweight rekeying strategy for effective key management and authentication during group member handoffs. This feature reduces the group key management overhead in wireless mobile networks where the group members are dynamically changing. The results of this research indicate that the DM-GKM framework is more cost-effective and robust compared with existing protocols.
- The DM-GKM framework solves the rekeying problem for both single-move and multi-moves across wireless networks while seamlessly participating in host mobility approaches. The DM-GKM framework solves the rekeying problem for both single-move and multi-moves across wireless networks while accounting for the host mobility scenario in wireless mobile environments. In addition, the DM-GKM framework minimizes the rekeying transmission, storage and communication overheads, optimizes the signaling load at the core network, eliminates single points of failure and prevents the 1-affects-*n* phenomenon.

- The BAN logic model has been used to analyze the mutual authentication verification of the DM-GKM framework and the results prove that the DM-GKM is a secure and reliable framework. The DM-GKM framework is proven to be robust against common security attacks.
- The simulation results indicate that the DM-GKM framework offers great flexibility, whereby the framework can be tuned to fulfil the requirements of a specific group based application and adapts automatically to account for group dynamism.

### 6.3 Future work

In general, the DM-GKM framework is a practical solution for group based applications in wireless mobile environments since it offers an efficient lightweight group key management for multiple groups of mobile users. This framework will indeed be useful for future wireless networks such as Software-Defined Networking (SDN) and Internet of Things (IoT). It is anticipated that the number of multicast services will escalate in the near future and consequently, the challenges of GKMPs will increase exponentially. Hence, future works should be focused on reducing the GKM overhead for large user groups in order to improve communication efficiency while simultaneously guaranteeing information security. It is recommended that the DM-GKM framework is implemented in real environment by means of a prototype. It is believed that the DM-GKM framework is a reliable solution for group based applications in order to provide efficient dynamic group services in real time environment. More importantly, the DM-GKM framework makes use of public key pairs with independent key per cluster which significantly reduces the computational overhead compared to symmetric encryption. Encryption is performed by the KDC which consumes more computational power only

when there is a membership change. Therefore, it is reasonable to assume that the encryption process will not reduce the performance of the DM-GKM.

## REFERENCES


Abdmeziem, M. R., Tandjaoui, D., & Romdhani, I. (2015). *A Decentralized Batch-based Group Key Management Protocol for Mobile Internet of Things (DBGK).* Paper presented at the Computer and Information Technology; Ubiquitous Computing and Communications; Dependable, Autonomic and Secure Computing; Pervasive Intelligence and Computing (CIT/IUCC/DASC/PICOM), 2015 IEEE International Conference on.

Adusumilli, P., Zou, X., & Ramamurthy, B. (2005). *DGKD: Distributed group key distribution with authentication capability.* Paper presented at the Proceedings from the Sixth Annual IEEE SMC Information Assurance Workshop.

Amir, Y., Kim, Y., Nita-Rotaru, C., Schultz, J. L., Stanton, J., & Tsudik, G. (2004). Secure group communication using robust contributory key agreement. *IEEE Transactions on Parallel and Distributed Systems, 15*(5), 468-480.

Anjum, F. (2006). *Location dependent key management using random key-predistribution in sensor networks.* Paper presented at the Proceedings of the 5th ACM workshop on Wireless security.

Ballardie, A. (1996). Scalable multicast key distribution.

Barrett, P. (1986). *Implementing the Rivest Shamir and Adleman public key encryption algorithm on a standard digital signal processor.* Paper presented at the Conference on the Theory and Application of Cryptographic Techniques.

Becker, K., & Wille, U. (1998). *Communication complexity of group key distribution.* Paper presented at the Proceedings of the 5th ACM conference on Computer and communications security.

Bellare, M., Desai, A., Jokipii, E., & Rogaway, P. (1997). *A concrete security treatment of symmetric encryption.* Paper presented at the Foundations of Computer Science, 1997. Proceedings., 38th Annual Symposium on.

Bouassida, M. S., Chrisment, I., & Festor, O. (2008). Group Key Management in MANETs. *IJ Network Security, 6*(1), 67-79.

Briscoe, B. (1999). *MARKS: Zero side effect multicast key management using arbitrarily revealed key sequences.* Paper presented at the International Workshop on Networked Group Communication.

Bruhadeshwar, B., & Kulkarni, S. S. (2011). Balancing revocation and storage trade-offs in secure group communication. *Dependable and Secure Computing, IEEE Transactions on, 8*(1), 58-73.

Burrows, M., Abadi, M., & Needham, R. M. (1989). *A logic of authentication.* Paper presented at the Proceedings of the Royal Society of London A: Mathematical, Physical and Engineering Sciences.

Camtepe, S. A., & Yener, B. (2004). *Combinatorial design of key distribution mechanisms for wireless sensor networks.* Paper presented at the European Symposium on Research in Computer Security.

Canetti, R., Garay, J., Itkis, G., Micciancio, D., Naor, M., & Pinkas, B. (1999). *Multicast security: A taxonomy and some efficient constructions.* Paper presented at the INFOCOM'99. Eighteenth Annual Joint Conference of the IEEE Computer and Communications Societies. Proceedings. IEEE.

Cao, J., Liao, L., & Wang, G. (2006). Scalable key management for secure multicast communication in the mobile environment. *Pervasive and Mobile Computing, 2*(2), 187-203.

Carneiro, G. (2010). *NS-3: Network simulator 3.* Paper presented at the UTM Lab Meeting April.

Chan, K.-C., & Chan, S.-H. (2002). Distributed servers approach for large-scale secure multicast. *IEEE Journal on selected areas in communications, 20*(8), 1500-1510.

Chang, B.-J., & Kuo, S.-L. (2009). Markov chain trust model for trust-value analysis and key management in distributed multicast MANETs. *IEEE Transactions on Vehicular Technology, 58*(4), 1846-1863.

Chen, I.-R., Cho, J.-H., & Wang, D.-C. (2006). *Performance characteristics of region-based group key management in mobile ad hoc networks.* Paper presented at the IEEE International Conference on Sensor Networks, Ubiquitous, and Trustworthy Computing (SUTC'06).

Chen, Y.-R., & Tzeng, W.-G. (2017). Group key management with efficient rekey mechanism: A Semi-Stateful approach for out-of-Synchronized members. *Computer Communications, 98*, 31-42.

Chiou, G.-h., & Chen, W.-T. (1989). Secure broadcasting using the secure lock. *IEEE Transactions on Software Engineering, 15*(8), 929-934.

Chu, H.-h., Qiao, L., Nahrstedt, K., Wang, H., & Jain, R. (2002). A secure multicast protocol with copyright protection. *ACM SIGCOMM Computer Communication Review, 32*(2), 42-60.

Daghighi, B., Kiah, M. L. M., Iqbal, S., Rehman, M. H. U., & Martin, K. Host mobility key management in dynamic secure group communication. *Wireless networks*, 1-19.

DeCleene, B., Dondeti, L., Griffin, S., Hardjono, T., Kiwior, D., Kurose, J., . . . Zhang, C. (2001). *Secure group communications for wireless networks.* Paper presented at the Military Communications Conference, 2001. MILCOM 2001. Communications for Network-Centric Operations: Creating the Information Force. IEEE.

Di Pietro, R., Mancini, L. V., & Jajodia, S. (2002). *Efficient and secure keys management for wireless mobile communications.* Paper presented at the Proceedings of the second ACM international workshop on Principles of mobile computing.

Diffie, W., & Hellman, M. (1976). New directions in cryptography. *IEEE transactions on Information Theory, 22*(6), 644-654.

Ding, C., Pei, D., & Salomaa, A. (1996). *Chinese remainder theorem: applications in computing, coding, cryptography*: World Scientific.

Dondeti, L., Mukherjee, S., & Samal, A. (1999). A distributed group key management scheme for secure many-to-many communication. *Department of Computer Science, University of Maryland, Tech. Rep. PINTL-TR-207-99*.

Dondeti, L. R., Mukherjee, S., & Samal, A. (2000). Scalable secure one-to-many group communication using dual encryption. *Computer Communications, 23*(17), 1681-1701.

Dutta, R., Mukhopadhyay, S., & Collier, M. (2010). Computationally secure self-healing key distribution with revocation in wireless ad hoc networks. *Ad Hoc Networks, 8*(6), 597-613.

Dutta, R., Mukhopadhyay, S., & Dowling, T. (2009). *Trade-off between collusion resistance and user life cycle in self-healing key distributions with t-revocation.* Paper presented at the Applications of Digital Information and Web Technologies, 2009. ICADIWT'09. Second International Conference on the.

Fujisaki, E., & Okamoto, T. (1999). *Secure integration of asymmetric and symmetric encryption schemes.* Paper presented at the Annual International Cryptology Conference.

Gharout, S., Bouabdallah, A., Challal, Y., & Achemlal, M. (2012). Adaptive Group Key Management Protocol for Wireless Communications. *J. UCS, 18*(6), 874-898.

Gharout, S., Bouabdallah, A., Kellil, M., & Challal, Y. (2010). *Key management with host mobility in dynamic groups.* Paper presented at the Proceedings of the 3rd international conference on Security of information and networks.

Gilks, W. R., Richardson, S., & Spiegelhalter, D. (1995). *Markov chain Monte Carlo in practice*: CRC press.

Goldwasser, S., Micali, S., & Rivest, R. L. (1988). A digital signature scheme secure against adaptive chosen-message attacks. *SIAM Journal on Computing, 17*(2), 281-308.

Gu, J., & Xue, Z. (2010). *An efficient self-healing key distribution with resistance to the collusion attack for wireless sensor networks.* Paper presented at the Communications (ICC), 2010 IEEE International Conference on.

Gu, X., Zhao, Y., & Yang, J. (2012). Reducing rekeying time using an integrated group key agreement scheme. *Communications and Networks, Journal of, 14*(4), 418-428.

Gudes, E. (1980). The design of a cryptography based secure file system. *IEEE Transactions on Software Engineering*(5), 411-420.

Hao, G., Vinodchandran, N., Ramamurthy, B., & Zou, X. (2005). *A balanced key tree approach for dynamic secure group communication.* Paper presented at the Proceedings. 14th International Conference on Computer Communications and Networks, 2005. ICCCN 2005.

Harney, H., & Harder, E. (1999). *Logical key hierarchy protocol*.

Harney, H., & Muckenhirn, C. (1997). *Group key management protocol (GKMP) architecture* (2070-1721).

Hernandez-Serrano, J., Pegueroles, J., & Soriano, M. (2005). *GKM over large MANET.* Paper presented at the Sixth International Conference on Software Engineering, Artificial Intelligence, Networking and Parallel/Distributed Computing and First ACIS International Workshop on Self-Assembling Wireless Network.

Huang, H.-F. (2009). A novel access control protocol for secure sensor networks. *Computer Standards & Interfaces, 31*(2), 272-276.

Index, C. V. N. (2013). Global mobile data traffic forecast update, 2012-2017. *Cisco white paper*.

Inoue, D., & Kuroda, M. (2004). *Fdlkh: Fully decentralized key management scheme on logical key hierarchy.* Paper presented at the International Conference on Applied Cryptography and Network Security.

Islam, S. H., Obaidat, M. S., Vijayakumar, P., Abdulhay, E., Li, F., & Reddy, M. K. C. (2017). A robust and efficient password-based conditional privacy preserving authentication and group-key agreement protocol for VANETs. *Future Generation Computer Systems*.

Jabeenbegum, S., Purusothaman, T., Karthi, M., Balachandar, N., & Arunkumar, N. (2010). *A cluster based cost effective contributory key agreement protocol for secure group communication.* Paper presented at the Computing Communication and Networking Technologies (ICCCNT), 2010 International Conference on.

Je, D.-H., Lee, J.-S., Park, Y., & Seo, S.-W. (2010). Computation-and-storage-efficient key tree management protocol for secure multicast communications. *Computer Communications, 33*(2), 136-148.

Jeng, F.-G., & Wang, C.-M. (2006). An efficient key-management scheme for hierarchical access control based on elliptic curve cryptosystem. *Journal of Systems and Software, 79*(8), 1161-1167.

John, S. P., & Samuel, P. (2010). *A distributed hierarchical key management scheme for mobile ad hoc networks.* Paper presented at the 2010 International Conference on Information, Networking and Automation (ICINA).

Kamat, S., Parimi, S., & Agrawal, D. P. (2003). *Reduction in control overhead for a secure, scalable framework for mobile multicast.* Paper presented at the Communications, 2003. ICC'03. IEEE International Conference on.

Kellil, M., Olivereau, A., & Janneteau, C. (2004). Rekeying in secure mobile multicast communications: Google Patents.

Kiah, M. L. M., & Martin, K. M. (2007). *Host mobility protocol for secure group communication in wireless mobile environments.* Paper presented at the Future Generation Communication and Networking (FGCN 2007).

Kim, Y., Perrig, A., & Tsudik, G. (2004). Tree-based group key agreement. *ACM Transactions on Information and System Security (TISSEC), 7*(1), 60-96.

Koo, H.-S., Kwon, O., & Ra, S.-W. (2009). A tree key graph design scheme for hierarchical multi-group access control. *IEEE Communications Letters, 13*(11), 874-876.

Kulkarni, S. S., & Bruhadeshwar, B. (2010). Key-update distribution in secure group communication. *Computer Communications, 33*(6), 689-705.

Kwak, D.-W., Lee, S. J., Kim, J. W., & Jung, E. (2006). An efficient LKH tree balancing algorithm for group key management. *IEEE Communications Letters, 10*(3), 222-224.

Kyntaja, T. (1995). A logic of authentication by Burrows, Abadi and Needham. *Science Helsinki University of Technology, Tehran. http://www. tml. tkk. fi/Opinnot/Tik-110.501/1995/ban. html*.

Lee, C.-C., Lin, T.-H., & Tsai, C.-S. (2009). A new authenticated group key agreement in a mobile environment. *annals of telecommunications-annales des télécommunications, 64*(11-12), 735-744.

Li, M., Poovendran, R., & Berenstein, C. (2002). Design of secure multicast key management schemes with communication budget constraint. *IEEE Communications Letters, 6*(3), 108-110.

Liao, L., & Manulis, M. (2007). Tree-based group key agreement framework for mobile ad-hoc networks. *Future Generation Computer Systems, 23*(6), 787-803.

Lin, C.-H. (1997). Dynamic key management schemes for access control in a hierarchy. *Computer Communications, 20*(15), 1381-1385.

Lin, J.-C., Huang, K.-H., Lai, F., & Lee, H.-C. (2009). Secure and efficient group key management with shared key derivation. *Computer Standards & Interfaces, 31*(1), 192-208.

Liu, D., & Ning, P. (2003). *Location-based pairwise key establishments for static sensor networks.* Paper presented at the Proceedings of the 1st ACM workshop on Security of ad hoc and sensor networks.

Liu, Y., Harn, L., & Chang, C. C. (2014). An authenticated group key distribution mechanism using theory of numbers. *International Journal of Communication Systems, 27*(11), 3502-3512.

Liu, Z., Ma, J., Huang, Q., & Moon, S. (2009). Asymmetric key pre-distribution scheme for sensor networks. *IEEE transactions on wireless communications, 8*(3), 1366-1372.

Lu, H. (2005). A novel high-order tree for secure multicast key management. *IEEE Transactions on Computers, 54*(2), 214-224.

Ma, D., Deng, R. H., Wu, Y., & Li, T. (2004). *Dynamic access control for multi-privileged group communications.* Paper presented at the International Conference on Information and Communications Security.

Madhusudhanan, B., Chitra, S., & Rajan, C. (2015). Mobility Based Key Management Technique for Multicast Security in Mobile Ad Hoc Networks. *The Scientific World Journal, 2015*.

Mapoka, T. T. (2013). Group key management protocols for secure mobile multicast communication: a comprehensive survey. *International Journal of Computer Applications, 84*(12).

Mapoka, T. T., Shepherd, S. J., & Abd-Alhameed, R. A. (2015). A new multiple service key management scheme for secure wireless mobile multicast. *IEEE Transactions on Mobile Computing, 14*(8), 1545-1559.

Mehdizadeh, A., Hashim, F., & Othman, M. (2014). Lightweight decentralized multicast–unicast key management method in wireless IPv6 networks. *Journal of Network and Computer Applications, 42*, 59-69.

Meyer, D. (1998). *Administratively scoped IP multicast* (2070-1721).

Mittra, S. (1997). *Iolus: A framework for scalable secure multicasting.* Paper presented at the ACM SIGCOMM Computer Communication Review.

Monahan, G. E. (1982). State of the art—a survey of partially observable Markov decision processes: theory, models, and algorithms. *Management Science, 28*(1), 1-16.

Mortazavi, S. A., Pour, A. N., & Kato, T. (2011). *An efficient distributed group key management using hierarchical approach with Diffie-Hellman and Symmetric Algorithm: DHSA.* Paper presented at the Computer Networks and Distributed Systems (CNDS), 2011 International Symposium on.

Munivel, E., & Lokesh, J. (2008). *Design of secure group key management scheme for multicast networks using number theory.* Paper presented at the Computational Intelligence for Modelling Control & Automation, 2008 International Conference on.

Nam, J., Lee, J., Kim, S., & Won, D. (2005). DDH-based group key agreement in a mobile environment. *Journal of Systems and Software, 78*(1), 73-83.

Naranjo, J. A. M., Antequera, N., Casado, L. G., & López-Ramos, J. A. (2012). A suite of algorithms for key distribution and authentication in centralized secure multicast environments. *Journal of Computational and Applied Mathematics, 236*(12), 3042-3051.

Nessett, D. M. (1990). A critique of the burrows, abadi and needham logic. *ACM SIGOPS Operating Systems Review, 24*(2), 35-38.

Ng, W. H. D., Howarth, M. P., Sun, Z., & Cruickshank, H. (2007). Dynamic balanced key tree management for secure multicast communications. *IEEE Transactions on Computers, 56*(5), 590-605.

Ng, W. H. D., & Sun, Z. (2005). *Multi-layers balanced LKH.* Paper presented at the IEEE International Conference on Communications, 2005. ICC 2005. 2005.

Nitaj, A. The Mathematical Cryptography of the RSA Cryptosystem.

Park, M.-H., Park, Y.-H., Jeong, H.-Y., & Seo, S.-W. (2013). Key management for multiple multicast groups in wireless networks. *IEEE Transactions on Mobile Computing, 12*(9), 1712-1723.

Park, M.-H., Park, Y.-H., & Seo, S.-W. (2010). *A cell-based decentralized key management scheme for secure multicast in mobile cellular networks.* Paper presented at the Vehicular Technology Conference (VTC 2010-Spring), 2010 IEEE 71st.

Pegueroles, J., & Rico-Novella, F. (2003). *Balanced batch LKH: new proposal, implementation and performance evaluation.* Paper presented at the Computers and Communication, 2003.(ISCC 2003). Proceedings. Eighth IEEE International Symposium on.

Penrig, A., Song, D., & Tygar, D. (2001). *ELK, a new protocol for efficient large-group key distribution.* Paper presented at the Security and Privacy, 2001. S&P 2001. Proceedings. 2001 IEEE Symposium on.

Piao, Y., Kim, J., Tariq, U., & Hong, M. (2013). Polynomial-based key management for secure intra-group and inter-group communication. *Computers & Mathematics with Applications, 65*(9), 1300-1309.

Quisquater, J.-J., & Couvreur, C. (1982). Fast decipherment algorithm for RSA public-key cryptosystem. *Electronics letters, 21*(18), 905-907.

Rafaeli, S., & Hutchison, D. (2002). *Hydra: A decentralised group key management.* Paper presented at the null.

Rivest, R. L., Shamir, A., & Adleman, L. (1978). A method for obtaining digital signatures and public-key cryptosystems. *Communications of the ACM, 21*(2), 120-126.

Rodeh, O., Birman, K., & Dolev, D. (1999). *Optimized group rekey for group communications systems*.

Roh, J.-H., & Lee, K.-H. (2006). *Key management scheme for providing the confidentiality in mobile multicast.* Paper presented at the 2006 8th International Conference Advanced Communication Technology.

Sanchez, D. S., & Baldus, H. (2005). *A deterministic pairwise key pre-distribution scheme for mobile sensor networks.* Paper presented at the First International

Conference on Security and Privacy for Emerging Areas in Communications Networks (SECURECOMM'05).

Scheikl, O., Lane, J., Boyer, R., & Eltoweissy, M. (2002). *Multi-level secure multicast: the rethinking of secure locks.* Paper presented at the Parallel Processing Workshops, 2002. Proceedings. International Conference on.

Setia, S., Koussih, S., Jajodia, S., & Harder, E. (2000). *Kronos: A scalable group re-keying approach for secure multicast.* Paper presented at the Security and Privacy, 2000. S&P 2000. Proceedings. 2000 IEEE Symposium on.

Shamir, A. (1979). How to share a secret. *Communications of the ACM, 22*(11), 612-613.

Sharma, S., & Krishna, C. R. (2015). *An efficient distributed group key management using hierarchical approach with elliptic curve cryptography.* Paper presented at the Computational Intelligence & Communication Technology (CICT), 2015 IEEE International Conference on.

Shen, V. R., & Chen, T.-S. (2002). A novel key management scheme based on discrete logarithms and polynomial interpolations. *Computers & Security, 21*(2), 164-171.

Sherman, A. T., & McGrew, D. A. (2003). Key establishment in large dynamic groups using one-way function trees. *IEEE Transactions on Software Engineering, 29*(5), 444-458.

Shinde, G., & Fadewar, H. (2008). *Faster RSA algorithm for decryption using Chinese remainder theorem.* Paper presented at the International Conference on Computational and Experimental Engineering and Sciences (ICCES).

Son, J.-H., Lee, J.-S., & Seo, S.-W. (2010). Topological key hierarchy for energy-efficient group key management in wireless sensor networks. *Wireless personal communications, 52*(2), 359-382.

Steiner, M., Tsudik, G., & Waidner, M. (1996). *Diffie-Hellman key distribution extended to group communication.* Paper presented at the Proceedings of the 3rd ACM conference on Computer and communications security.

Steiner, M., Tsudik, G., & Waidner, M. (1997). *CLIQUES: A new approach to group key agreement*: Citeseer.

Striki, M., Baras, J. S., & Manousakis, K. (2006). *A robust, distributed TGDH-based scheme for secure group communications in MANET.* Paper presented at the 2006 IEEE International Conference on Communications.

Sun, H.-M., He, B.-Z., Chen, C.-M., Wu, T.-Y., Lin, C.-H., & Wang, H. (2015). A provable authenticated group key agreement protocol for mobile environment. *Information Sciences, 321*, 224-237.

Sun, Y., & Liu, K. (2007). Hierarchical group access control for secure multicast communications. *IEEE/ACM Transactions on Networking (TON), 15*(6), 1514-1526.

Sun, Y., & Liu, K. R. (2004). *Scalable hierarchical access control in secure group communications.* Paper presented at the INFOCOM 2004. Twenty-third AnnualJoint Conference of the IEEE Computer and Communications Societies.

Sun, Y., & Liu, K. R. (2007). Hierarchical group access control for secure multicast communications. *IEEE/ACM Transactions on Networking, 15*(6), 1514-1526.

Sun, Y., Trappe, W., & Liu, K. R. (2004). A scalable multicast key management scheme for heterogeneous wireless networks. *IEEE/ACM Transactions on Networking (TON), 12*(4), 653-666.

Thangavel, M., Varalakshmi, P., Murrali, M., & Nithya, K. (2015). An Enhanced and Secured RSA Key Generation Scheme (ESRKGS). *Journal of Information Security and Applications, 20*, 3-10.

Thomas, R., & Salim, A. (2014). *A novel decentralized group key management using attribute based encryption.* Paper presented at the Computational Systems and Communications (ICCSC), 2014 First International Conference on.

Trappe, W., Song, J., Poovendran, R., & Liu, K. R. (2003). Key management and distribution for secure multimedia multicast. *IEEE Transactions on Multimedia, 5*(4), 544-557.

Tsitsipis, D., Tzes, A., & Koubias, S. (2014). Talk: topology aware LKH key management. *International Journal of Distributed Sensor Networks, 2014*.

Van Tilborg, H. C., & Jajodia, S. (2014). *Encyclopedia of cryptography and security*: Springer Science & Business Media.

Varalakshmi, R., & Uthariaraj, V. R. (2014). Huddle hierarchy based group key management protocol using gray code. *Wireless networks, 20*(4), 695-704.

Veltri, L., Cirani, S., Busanelli, S., & Ferrari, G. (2013). A novel batch-based group key management protocol applied to the internet of things. *Ad Hoc Networks, 11*(8), 2724-2737.

Vijayakumar, P., Bose, S., & Kannan, A. (2014). Chinese remainder Theorem based centralised group key management for secure multicast communication. *IET information Security, 8*(3), 179-187.

Waldvogel, M., Caronni, G., Sun, D., Weiler, N., & Plattner, B. (1999). The VersaKey framework: Versatile group key management. *IEEE Journal on selected areas in communications, 17*(9), 1614-1631.

Wallner, D., Harder, E., & Agee, R. (1999). *Key management for multicast: Issues and architectures* (2070-1721).

Wan, Z., Ren, K., Lou, W., & Preneel, B. (2008). *Anonymous id-based group key agreement for wireless networks.* Paper presented at the 2008 IEEE Wireless Communications and Networking Conference.

Wang, F., Yu, F. R., & Srinivasan, A. (2009). *Distributed hierarchical key management scheme in mobile ad hoc networks.* Paper presented at the Global Telecommunications Conference, 2009. GLOBECOM 2009. IEEE.

Wang, Y., Li, J., Tie, L., & Zhu, H. (2004). *An efficient method of group rekeying for multicast communication.* Paper presented at the Emerging Technologies: Frontiers of Mobile and Wireless Communication, 2004. Proceedings of the IEEE 6th Circuits and Systems Symposium on.

Weiler, N. (2001). *SEMSOMM-A scalable multiple encryption scheme for one-to-many multicast.* Paper presented at the Enabling Technologies: Infrastructure for Collaborative Enterprises, 2001. WET ICE 2001. Proceedings. Tenth IEEE International Workshops on.

Wong, C. K., Gouda, M., & Lam, S. S. (2000). Secure group communications using key graphs. *IEEE/ACM transactions on networking, 8*(1), 16-30.

Wu, T.-Y., Tseng, Y.-M., & Yu, C.-W. (2011). A Secure ID-Based Authenticated Group Key Exchange Protocol Resistant to Insider Attacks. *J. Inf. Sci. Eng., 27*(3), 915-932.

Xu, G., Chen, X., & Du, X. (2012). *Chinese Remainder Theorem based DTN group key management.* Paper presented at the Communication Technology (ICCT), 2012 IEEE 14th International Conference on.

Xu, S., Yang, Z., Tan, Y., Liu, W., & Sesay, S. (2005). *An efficient batch rekeying scheme based on one-way function tree.* Paper presented at the IEEE International Symposium on Communications and Information Technology, 2005. ISCIT 2005.

Xu, Y., Zhou, W., & Wang, G.-J. (2014). Multiway tree-based group key management using Chinese remainder theorem for multi-privileged group communications. *淡江理工學刊, 17*(1), 81-92.

Yan, J., Ma, J., & Liu, H. (2009). Key hierarchies for hierarchical access control in secure group communications. *Computer Networks, 53*(3), 353-364.

Yang, C., & Li, C. (2004). Access control in a hierarchy using one-way hash functions. *Computers & Security, 23*(8), 659-664.

Younis, M., Ghumman, K., & Eltoweissy, M. (2005). *Key management in wireless ad hoc networks: collusion analysis and prevention.* Paper presented at the PCCC 2005. 24th IEEE International Performance, Computing, and Communications Conference, 2005.

Younis, M. F., Ghumman, K., & Eltoweissy, M. (2006). Location-aware combinatorial key management scheme for clustered sensor networks. *IEEE Transactions on Parallel and Distributed Systems, 17*(8), 865-882.

Yu, F. R., Tang, H., Mason, P. C., & Wang, F. (2010). A hierarchical identity based key management scheme in tactical mobile ad hoc networks. *IEEE Transactions on Network and Service Management, 7*(4), 258-267.

Zhang, C., DeCleene, B., Kurose, J., & Towsley, D. (2002). Comparison of inter-area rekeying algorithms for secure wireless group communications. *Performance Evaluation, 49*(1), 1-20.

Zhang, Q., & Calvert, K. L. (2003). *On rekey policies for secure group applications.* Paper presented at the Computer Communications and Networks, 2003. ICCCN 2003. Proceedings. The 12th International Conference on.

Zhang, Q., & Wang, Y. (2004). *A centralized key management scheme for hierarchical access control.* Paper presented at the Global Telecommunications Conference, 2004. GLOBECOM'04. IEEE.

Zhao, S., Kent, R., & Aggarwal, A. (2013). A key management and secure routing integrated framework for Mobile Ad-hoc Networks. *Ad Hoc Networks, 11*(3), 1046-1061.

Zheng, X., Huang, C.-T., & Matthews, M. (2007). *Chinese remainder theorem based group key management.* Paper presented at the Proceedings of the 45th annual southeast regional conference.

Zheng, X., Manton, M., & Huang, C.-T. (2007). *Hierarchical scalable group key management based on Chinese remainder theorem.* Paper presented at the Proceedings of the 6th Annual Security Conference, Apr.

Zhou, J., & Ou, Y. H. (2009). Key tree and Chinese remainder theorem based group key distrubution scheme. *Journal of the chinese institute of engineers, 32*(7), 967-974.

Zhu, W. T. (2005). Optimizing the tree structure in secure multicast key management. *IEEE Communications Letters, 9*(5), 477-479.