# An Implementation of Secure Group Communication in a Wireless Environment

Miss Laiha Mat Kiah and Babak Daghighi

***Abstract*—In recent years, group based applications and protocols have gradually gained popularity in term of data traffic distribution across a group of members. Since these applications typically involve communication over open networks, security has become an important requirement. Many group key management schemes have been suggested for provision of secrecy in a group communication. Nonetheless, little implementations have been carried out in actual wireless environment. This paper is focused to deploy a decentralized group key management scheme using Java 2 Mobile Edition (J2ME) platform for mobile devices and Java 2 Standard Edition (J2SE) platform for enabling the communication between key management servers and mobile devices. To evaluate the accuracy of the scheme, different aspects of it in particular the join and leave operations including updating key materials are then tested and discussed.**

***Index Terms*—Secure group communication, group key management, and mobile environment**

## I. Introduction

Ensuring the security of group based applications such as Tele/video -conferencing, stock updates as well as social group networks is no trivial matter since most of group based applications take place over insecure network. Depending on the application need, basic security services such as confidentiality, data integrity and entity authentication need to be in place to ensure backward and forward secrecy, as well as the integrity of group members and group operations. These services in particular the backward and forward secrecy can be established by sharing a common key, which then used to encrypt all traffic of a particular group. Only members of the group can decrypt the received message. Managing a group key is one of fundamental challenges in designing secure and reliable group communication system. There are several group key management schemes for disseminating group key to members of a group, which can be classified into three categories: 1) Centralized 2) Distributed 3) Contributory.

Centralized group key management involves a single entity as a group controller responsible for the generation, distribution and updating the group key. Logical Key Hierarchy approach is one of the famous schemes in this category that was proposed by several research groups nearly at the same time [1]-[2]. Other existing approach can be found in [3]-[8].

Manuscript received October 14, 2012; revised November 15, 2012. This work was supported in part by the University of Malaya under Grant UMRG–RG029/09ICT.

The authors are with the Faculty of Computer Science & IT, University of Malaya, and KL 50603 Malaysia (e-mail: misslaiha@um.edu.my, Babak@um.edu.my).

Although adopting centralized approach have some advantages such as 1) easy management because the provision of trust is focused on one entity and 2) some transmission overheads are decreased as member of group need to authenticate the main entity only one time, they suffer from some weaknesses as follows: 1) dependencies on a key server leave a single point of failure, 2) also, it must be constantly available during group operations, and 3) for larger group size, the amount of message transmission between the key manager and group members can be dramatically increased at a same time, which could create bottleneck.

In distributed or contributory approach, there is no explicit key entity or centre, and all members contribute to managing the key(s). This approach eliminates the need for central entity while providing uniform distribution of the work load for key management. While it alleviates the single point of failure problem in centralized group key management; it still suffers from some drawbacks as follows: 1) distribution of management task across large multicast groups is complicated, and 2) the processing time and communication requirement are increased along with the size of group members. Several schemes on distributed approach have been presented in [9]-[12].

In contrast with the discussed approaches, decentralized group key management scheme can be a combination of former approaches (centralized and distributed). A large group is split to smaller subgroups placed in hierarchical levels. Each level which could consist of one or more entities is responsible for key management in its level, while maintaining some dependencies on the upper level entity. Such scheme was first proposed by IOLUS [13] followed by some improvements schemes such as in [14]-[16]. We observed that this scheme is more desirable for employing in this work.

The aim of this paper is to implement a group key management scheme in wireless environment. Meanwhile, the main components of scheme including entities and protocols are identified for providing a secure group communication. The implementation of the proposed scheme is presented. Moreover, the analysis and discussion of the implemented scheme is given.

## II. The Proposed Scheme

According to [15] and [17], the main elements of decentralized group key management scheme are group manager which is responsible to govern all group processes, and group members. These entities were placed in hierarchy levels, and a traffic key is shared between members of a group which is typically used for encrypting

the data traffic. Adopting the domain and area aspects of the existing scheme, our proposed scheme is as follows: The main components of the scheme consist of key managers and protocols (i.e. governing the group operations/processes). The key managers are comprised of the domain key manager and the area key managers. The main protocols involved are the join, leave and rekeying protocols.

The roles and responsibilities of these components are as follows:

- Domain Key Manager (DKM): responsible for generating, distributing, storing and deleting all keying materials of a domain.
- Area Key Manager (AKM): under *DKM*'s authority, unique per area responsible for key management of an area.
- Group Member (M): any node who wishes to participate in a group communication.

The placement of entities is illustrated in Fig.1. The entity which is labeled by *DKM* in domain Z is domain key manager. Within areas *a* and *b*, the entity labeled with *AKM* is area key manager. The members of a group are placed in the different areas within the domain Z.

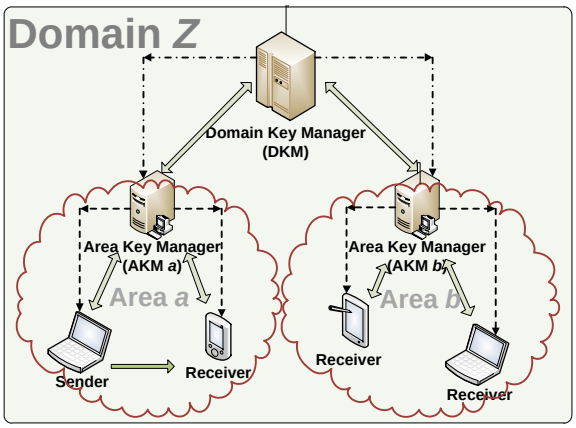


Fig. 1. An example of placement of entities.

### A. Type of Keys

A variety of keys are used by entities when they want to initiate a secure group communication. The proposed scheme used symmetric key approach which is suitable for wireless environment due to the restrictions in terms of power and computation [18] that most wireless nodes exhibit. Type of keys involved in this scheme is as follows:

- Domain-Area keys (DAi_Key): unique, shared between the DKM and AKMi of area i.
- Domain Key (D_Key): generated by *DKM*, shared with all *AKM*s in a domain.
- Area-Member Key ($A_i$M_Key): unique area key, shared between AKM and group member *M* in an area.
- Area Key (A_Key): generated by *AKM*, shared with all group members in an area.
- Group-Traffic Keys (T_Key): shared between all members of a group in a domain for securing data traffic.

### B. Protocol Functionalities

The functionalities of protocols in this proposed scheme are described as follows. Note that message exchanges between member and area key manager *i* are encrypted/decrypted under $A_i$*M_Key* key. As such, *DKM* encrypts/decrypts the messages exchanged between itself and the area key manager *i* under *DAi_Key* key.

*Protocol I: New Member Joining to a Group Communication.* This protocol manages join operation of a new host member to a group. In order to preserve backward secrecy (so that the previous data traffic is not accessible to the new member), the generation of the new traffic key *T_Key* occurs. At the end of the join protocol, the new member obtains the current set of cryptographic key, *T_Key* and *A_Key*.

*Protocol II: Existing Member Leaving the Group Communication.* This protocol manages leave operation when an existing group member departs a group with consideration no access to future data traffic i.e. the provision of forward secrecy. For that reason, the generation and distribution of new traffic key has to occur. At the end of the leave protocol, all group members excluding the departing member obtain a new set of keying material.

*Protocol III: Rekeying Group-Traffic Key.* Rekeying protocol is employed for provision of backward secrecy in member joining as well as forward secrecy in member leaving a group. Note that rekeying of group traffic key always run immediately upon any membership changes (during joining and/or leaving operation).

## III. Implementation

The implementation of the proposed scheme is done with three main entities; a domain key manager (DKM), an area key manager (AKM) and group members (M). While DKM and AKM entities can reside on the same device, group members can be a device with wireless network interface card. While Java 2 Mobile Edition (J2ME) is used for the development on mobile devices, Java 2 Standard Edition (J2SE) is used to enable the communication with the key management servers, and to develop the required functions as discussed above. The implementation is presented by looking at each of main entities. There are three main entities:

### A. Domain Key Manager (DKM)

*DKM* listens to requests received through the established connection with the *AKM*. To do this, *DKM* creates a server socket object to a specific port. The *DKM* entertains an *AKM*'s request by executing the area key manager handler class. Same class also manages the rekeying protocol at the domain level.

### B. Area Key Manager (AKM)

The implementation of *AKM* is done in two aspects; client and server respectively. That means *AKM* acts as a client for *DKM* while a server for group members in its area. The implementation of *AKM* consists of three distinctive classes; registration to *DKM* server, client handler and *DKM* message handler. The registration to *DKM* server is done by *AKM* via a socket object by attaching *DKM*'s IP address and its port. As client

handler, *AKM* acts as server for receiving new connection/requests from group members residing in its area. Finally, acting as client to *DKM*, via *DKM* message handler class, *AKM* continuously listens to *DKM* server for receiving *DKM* server commands which can include any notification and control messages for group operations.

### C. Group Member (M)

The implementation of a group member is done simply by sending its requests to *AKM* server to obtain all the keying materials prior to its participation in a group. For example, to join a group, *M* sends the *join_request* message to the *AKM* server at which the join request is processed.

### D. Implementation on Real Mobile Device

To demonstrate further, actual implementation of group member's operation is done on real mobile device i.e. mobile phone. As mentioned, *DKM* and *AKM* entities can be implemented on separate or same device. In this, we implement both entities on the same device. Three different ways can be used to connect the mobile phone with the *AKM*; via SMS, via the Internet, and via wireless access points. We chose to use the SMS means as that is the cheapest solution and can be used by most mobile phones available at present.

The implementation is divided in two parts; server side and client side. On the server side, we developed the required classes that enable *AKM* to read/send the SMS from/to the mobile phone via the GSM modem. On the client side, we developed the required software to enable the mobile phone to communicate with the *AKM*. Fig. 2 shows the *AKM* while receiving a request from the mobile phone client. The implementation of mobile phone client part is shown in Fig. 3.

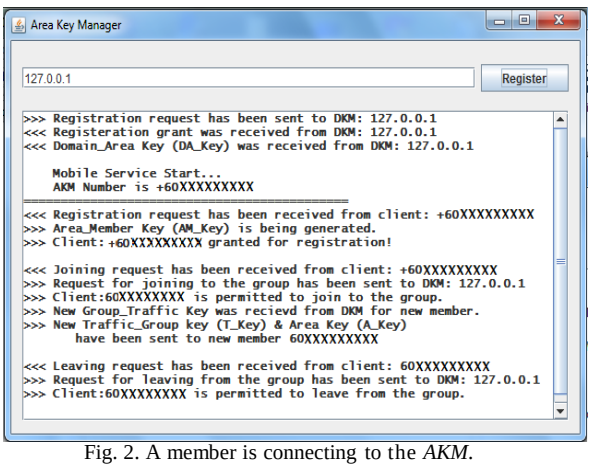


Fig. 2. A member is connecting to the *AKM*.

## IV. Analysis and Discussion

From the implemented proposed scheme, if a member, *M* is interested in joining a multicast group such as 224.0.0.1, it sends a join request message to the area key manager, *AKM*. *AKM* verifies the request and sends it to the domain key manager, *DKM*. *DKM* then extracts the IP address of desired multicast group. Assuming that *M* is the first member of the multicast group, the *DKM* generates the traffic key *T_Key* and sends it to the *AKM* who then delivers the key to *M*. As described above, this operation satisfies Protocol I.

In case of an existing member, *M* wishes to leave the multicast group, *M* sends a leave notify message to its *AKM* who then in turn sends the request to the *DKM*. Upon receiving the request, *DKM* proceeds to process the leaving member, and initiates the re-keying (i.e. updating of keying material) operation. This satisfies Protocol II as described above.


Fig. 3. Mobile phone client - software screen

Our proposed scheme carried out the updating of key material in two occurrences; upon joining and leaving operations of member, *M*. Each time, the *DKM* initiates the re-keying operation via ready to rekey message to the *AKM*. As a result, a new traffic key is delivered to *M*. The re-keying of new traffic key is immediately initiated upon changes in group membership (i.e. due to new member joining or existing member leaving) for the provision of backward and forward secrecy. As described above, the re-keying protocol is satisfied.

## V. Conclusion

Group key management is fundamental building block for provision of security in group based application, which can be provided through the implementation of appropriate cryptographic key mechanisms. Since wireless networks have been increasingly prevailing, existing applications in wired environment should be made available in wireless environment. The overall aim of this research was to deploy a key management for secure group communication in wireless environment.

Base on decentralized approach specific for wireless environment, we identified main components and functionalities necessary for establishing a key management scheme. Detailed discussions on key management framework including its main components and protocols have been provided. The implementation of the proposed scheme has been discussed and analyzed accordingly. We have shown that our group key management scheme met the requirements for a secure group communication. We note that further works are being carried out in regards to host mobility issues involving the placement of group members within two or more areas.

## Acknowledgment

This research has been funded from University of Malaya under grant UMRG–RG029/09ICT.

## REFERENCES


[1] C. K. Wong and M. Gouda, and S. S. Lam, *Secure group communications using key graphs*, IEEE/ACM Transactions, 1997, pp. 16-30.

[2] D. Wallner, E. Harder, and R. Agee, *Key management for multicast: Issues and architectures*, RFC 2627, 1999.

[3] N. Shang *et al*., "A privacy-preserving approach to policy-based content dissemination," Data Engineering (ICDE), in *Proc. of 2010 IEEE 26th International Conference on*, pp. 944-955.

[4] M. Blanton and K. Frikken, "Efficient multi-dimensional key management in broadcast services," *Computer Security–ESORICS 2010*, 2011, pp. 424-440.

[5] C. Zheng, Y. Pengpeng, and L. Zhengjun, "A Key Management Scheme for Multicast Based on Layer 2 Controls," in *Proc. of IEEE 10th International Conference*, pp. 1512-1518, 2010.

[6] R. Canetti *et al*., "Multicast security: taxonomy and some efficient constructions," in *Proc. of IEEE Conference*, pp. 708-716, 1999.

[7] M. Li *et al*., "Approximately optimal trees for group key management with batch updates," *Theoretical Computer Science*, 2009, vol. 410, no. 11, pp. 1013-1021.

[8] A. Aikebaier and M. Takizawa, "A protocol for reliably, flexibly, and efficiently making agreement among peers," *International Journal of Web and Grid Services*, 2009, vol. 5, no. 4, pp. 356-371.

[9] B. Daghighi, M. L. M. Kiah, and S. Afkhami, "Group Key Management Using Public Key Exchange," in *Proc. of International Conference on Intelligent network & Computing*, 2010.

[10] M. Abdalla, *et al*., "Flexible group key exchange with on-demand computation of subgroup keys," *progress in cryptology–Africa Crypt*, pp. 351-368, 2010.

[11] Y. Kim, A. Perrig, and G. Tsudik, *Simple and fault-tolerant key agreement for dynamic collaborative groups*, 2000, ACM.

[12] I. A. Saroit, S. F. El-Zoghdy, and M. Matar, "A Scalable and Distributed Security Protocol for Multicast Communications," in *Proc. of International Journal of Network Security*, vol. 12, no. 2, pp. 61–74, 2011.

[13] K. Becker and U. Wille, *Communication complexity of group key distribution*, ACM, 1998.

[14] L. R. Dondeti, A. Samal, and S. Mukherjee, "A dual encryption protocol for scalable secure multicasting," *IEEE Computer Society*, 1999.

[15] B. Briscoe, "Zero side effect multicast key management using arbitrarily revealed key sequences," *Networked Group Communication*, 2004, pp. 301-320.

[16] J. H. Cho, I. R. Chen, and M. Eltoweissy, "On optimal batch rekeying for secure group communications in wireless networks," *Wireless Networks*, vol. 14, no. 6, pp. 915-927, 2008.

[17] L. Kiah and K. M. Martin, "Host Mobility Protocol for Secure Group Communication in Wireless Mobile Environments," *IEEE*, 2008.

[18] M. L. M. Kiah, "A Key Management Framework for Secure Group Communication in Wireless Mobile Environments," Department of Mathematics, University of London.