Group Key Management in Wireless Networks Using Session Keys

Dhilak Damodaran, Rohit Singh, Phu Dung Le
Peninsula School of Information Technology, Monash University
McMahons Road, Frankston, Victoria 3199, Australia

Abstract

Recent developments in the field of wireless networks have led to the phase where it has now become important to offer the members connected in a wireless network with a secure and efficient group key management system for accessing numerous services. A secure group key agreement protocol designed for multicast communications using short life span session keys [5] is proposed in this paper. Our proposed work guarantees key and data authenticity among the group members, and data confidentiality of group messages in addition to the properties of forward and backward secrecy. The group key is collaboratively established by combining the keys of all authenticated members, which helps in maintaining the communication and computation transparency among the group members.

Keywords: *Wireless, Portable, Group Key, Session Keys, Multicast, Forward and Backward Secrecy.*

1. Introduction

There has been an inimitable growth of Internet over the last few years coupled with the increase of bandwidth in present day networks. These dynamic and rapid changes have resulted in the development of many emerging multicast services all of which have led to the swift progress of group communication model.

Various schemes [3] [4] [6] [10] have been proposed for the secure distribution of group key to group members in a wired network. Ideally, group communications must also be applied to wireless environment. Therefore it has become quite demanding to forecast the combination of wireless and multicast services which will lead to the development of several multicast applications and services running on the wireless platform.

The main focus in designing a key management scheme for wireless networks is to back it up with a strong security feature amongst the group members. The vital prerequisite for secure group communications is to generate a common group key for all members to securely communicate with each other. In addition to the issue of secure group communication model, the issues related to the efficient distribution of data among a group of users have also been dealt in the literature. Yet by far, there is still not a very clear understanding about the technique that describes the use of session keys in a multicast session in wireless networks.

The lack of security is predominantly due to poor management of keys rather than the weakness in the encryption algorithm itself [7]. Lending weight to this factor we have proposed the concept of generating a session key for secure group communication because even the strongest of the encryption technique is vulnerable to attacks if the key of a member within the group gets compromised. The concept of session keys is used in the proposed protocol for mutual authentication and group key generation, hence key agreement can be efficiently achieved without compromising the security.

This paper is organized as follows: The standard framework model of a group communication system is discussed in Section 2. Section 3 describes our proposed wireless group communication model using session keys. Section 4 analyses the rekeying issues involved in secure group communications, followed by the discussion on security issues in Section 5 of our proposed work. Section 6 concludes our work.

2. Standard Framework Model

In this section we describe the general framework model that is used for group communications model in a wireless network, which also forms the basis for our proposal.

2.1. Mobility Framework Model

The lack of security in the group communication model hampers the effective and large scale deployment of various tactical multi-user applications. The issues involved in multicast security are [6]: data confidentiality, group key management, data source authentication, and security policies. The IETF multicast security working group [10] has also identified three problem areas, viz., key distribution, data origin authentication, and policy management, in secure group communication.

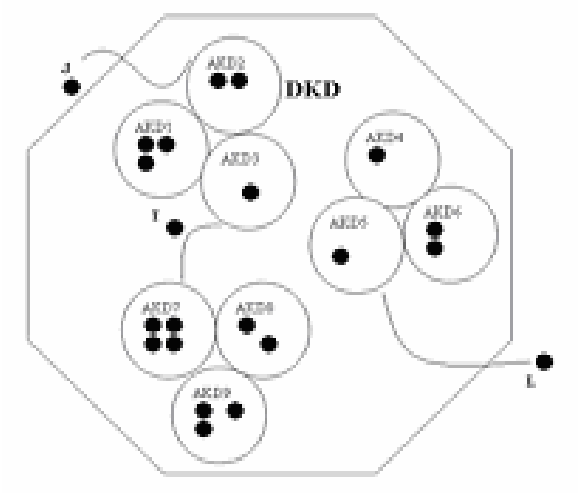
Group communications model is much more complex and difficult to comprehend when applied to the wireless networks due to their mobility factor. To make sure that only the authenticated members will be able to access the

Proceedings of the Third International Conference on Information Technology: New Generations (ITNG'06)
0-7695-2497-4/06 $20.00 © 2006 IEEE

information during the period corresponding to their authorization, a *Group Key (GK)* is used to encrypt group data. With the GK the key server can ensure that the data can be decrypted only by the members of the group.

The mobility of members in the wireless networks can be explained by following the mobility framework model [2] as shown in Figure 1. All the members in the group who are sparsely located in various cells come under a common location called the "*domain*". The domain is denoted by the assortment of circles, and is managed by a *Domain Key Distributor* (DKD) [4]. The independent "cells" that make up the domain are called *"areas"* and are managed by an *Area Key Distributor* (AKD) [4].

The DKD is responsible for the generating and distributing the group key to all the AKD's, which in turn will distribute them to the local members present within each of their own areas. As the distribution of the Data Encryption Key (DEK) has to be secure, it is protected by a local *key encryption key* (KEK). When a member moves within the cell of its locality it does not initiate any rekeying with the join/leave operations leveraged within the area by an *intraarea* rekeying algorithm [12]. Given a situation that a member needs to move between two different cells then the *interarea* rekeying algorithms [12] influences the re-grouping of the members into the newly transferred area.



Index: J – member join, L – member leave, T – member transfer

Figure 1.Mobility framework model

2.2. Logical key tree structure

In real time scenario, a group is commonly divided into fraction of several smaller subgroups, for instance, a multicast group of hospital can consist of hospital branches, staff id's, and client id's as their subgroups. Every member within the group will be able to communicate with other members belonging to the group, and also participate in the group communications. Presently, most of the group key management algorithms use hierarchical structure (tree structure) to describe the logical keying mechanisms. Logical Key Hierarchy (LKH) is a mechanism for secure key management within a group of entities providing the ability to initialize the group with a common key and then to rekey the group as required [12]. The LKH entity logical key tree structure shown in Figure 2 consists of three types of entities: a Key Server (KS), Subgroup Keys (K_n), and one or more group members (U_n). The key server is responsible for generating, allocating, and rekeying (to maintain security) the group key as appropriate; and the group members are entities with access to the group keys. The LKH scheme has several advantages like scalability, collusion-proof, independent of any encryption or decryption algorithm to name a few.

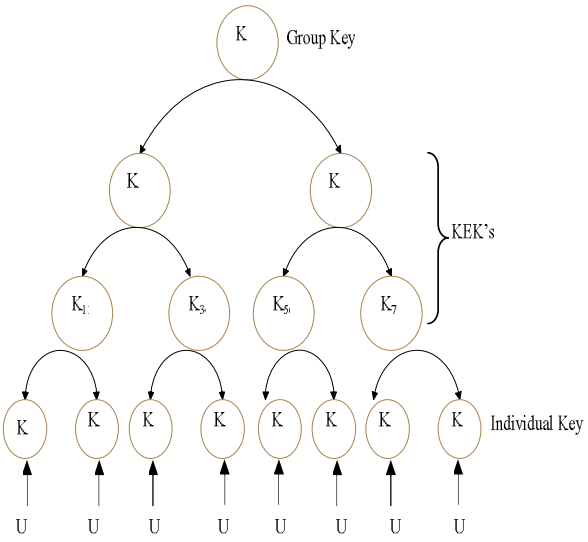


Figure 2.Logical key tree structure

3. Our Proposed Wireless Group Communication Model

Our contribution in this paper is a provably-secure group key agreement protocol for wireless networks based on the generation of session keys.

3.1. Notations Used

- *{C, A, S}*: the set of members, access point and server, respectively.
- *MID*: member's user id.
- *CI*: Command execution information.
- *CRes*: Command response
- $\{M\}_X$: the message M symmetrically encrypted with the shared key *X*.

- *{M} Pub_X*: the message M encrypted with the public key of the party *X*.
- *{M}$Priv_X$*: the message M signed with the private key of the party *X*.
- *h (M)*: the one-way hash function of the message *M*.
- *MAC (M, K)*: the message authentication code (*MAC)* of the message *M* with the key *K*.
- *SK, IK, GK*: Session, Individual and Group Key.

3.2. Initial Assumptions and settings

1. The distributed key (DK) is a key shared between member and server. This key is further used to generate session key Y_i. Member's user id (MID), containing the member's id and other information about the member is the long time shared secret between the server and member. This data is distributed by performing Authentication Key Exchange (AKE) protocol [14] between the member and server.

$$\mathbf{C \rightarrow S}: \{MID, DK, n\}_k$$
$$\mathbf{S \rightarrow C}: \{n\}_k$$

2. The session key is generated by the following procedure:

$$Y_1 = h(SIK, DK), Y_2 = h(SIK, Y_1), Y_n = h(SIK, Y_{n-1})$$

where, SIK is the session initialization key.

3. Every member generates their Individual Key at the terminal and transmits it to the respective Access Point, which forwards the same to the server. The procedure for generating the IK is as follows:

$$C_n \text{ performs} \rightarrow \mathbf{HMAC\ (MID, Y_i) \rightarrow IK_n}$$

The IK of every member is also computed on the server side as well. If the value computed matches the value received, then the member is authenticated to join the group. This step corresponds to the end of step 1 of the first tier in the group communication model as depicted in figure 3.

3.3. Group Communication Model Architecture

We present a new group key management system, which can significantly improve the security of key management in wireless networks. We describe the proposed system from the perspective of system architecture that can be used for banking system, medical databases, universities, and so on.

In the proposed system architecture, we adopt the wireless member-server model and extend it for the group communication model in wireless networks. The whole wireless domain is split into several administrative fractions which in turn are based on different areas. The entire architecture is subdivided into two tiers comprising of the following five step process.

First-tier:

Step 1: It is in this tier where the authenticity of the member who wants to join the group is decided. If the member proves to be a legitimate user then s/he is allowed to join the group, otherwise, a denial response is sent back to the user.

Second-tier:

Step 2: After successfully authenticated by the server the new member is clustered along with other authenticated members of the group. Once the grouping is completed the multicast session begins. During the multicast session each member of the group contributes their share to form the group key which is performed by the server and distributed to the members through the access point (A).

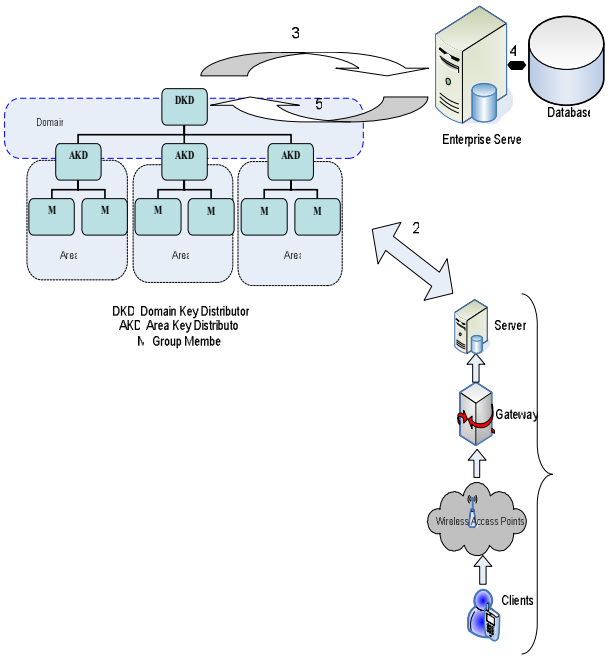


Figure 3.Wireless group communication model

Step 3: When a member wants to utilize the services from the database, s/he forwards the requests for connection with the enterprise server using the individual key (known only to the member and the server), and the group key common to the group.

Step 4: On receiving the connection request the enterprise server checks for the authenticity of the requesting members and if found to be legitimate, then collects the necessary information from the database and returns the equivalent response.

Step 5: The response from the enterprise server is then sent back to the member.

3.4. Applying the Proposed Technique

Step 1: $C_n \rightarrow$ A: ID_C, IK, r,

where IK= [HMAC (MID, Yi)]

A → S: IK, r

Step 2: $\sum_{i=1}^{n}$

S →*A:* GK = IK

A → C_n: GK

Step 3: **C→ S:** MAC [(IK, GK, CI), Yi], CI, IK, GK

Step 4: **S→C:** $\{CRes\}_{Yi}$

The command information (CI) and command response (CRes) vary according to the different information requested by different members.

Step 1: Members compute their IK and transmits IK, ID and r (random number to compute the set of generate session keys [7]) to the access point, which is then sent to server.

Step 2: Upon successful authentication of the member, the Server allows the member to join the group. This process is repeated for all the members who want to join the group. Once the grouping of all the members is complete the server computes the GK. The computed value is then sent back to the access points which redistributes it back to the respective members.

Step 3: If the member wants to retrieve any information from the database he can do so by sending in the requested query along with the individual and group keys. This allows the server to be sure that the request is indeed from a legitimate member.

Step 4: The server then computes the MAC value of (IK,GK,CI), and if the result matches with the value sent by the member it then sends back the information requested by the member.

4. Rekeying Issues

The aforementioned details explained how the session key can be effectively used for the generation of individual and group key eventually leading to the multicast session. However, in a multicast group addition and deletion of members are quiet possible while the session is in progress, which results in a phase called rekeying. As a result, the rekeying issues are also considered in our protocol. The main feature that makes the rekeying process more efficient is the fact that each member knows only the set of keys along the path from its base node up to the root. When membership changes such as addition and deletion of members take place, the rekeying procedure is invoked to update the affected keys thereby ensuring security. When a new member joins the group, the group key needs to be updated in order to prevent the new member from accessing the past communication details. This property is known as backward access control [1] [4] [8]. Likewise, when a member is deleted or removed from the group, it again becomes necessary to change the common group key in order to prevent the leaving member from gaining access to the future communication details. This property is known as forward access control [1] [4] [8]. Changing the group key after a join operation is easy (GK can be sent via multicast to all the members including the new member) when compared to the delete operation. Therefore, the guarantee of maintaining the confidentiality of group communication is rather a tough task to resolve while considering large and dynamic groups because of difficulties in executing the persistent change of keys (both individual and group) whilst maintaining low computation, bandwidth and storage overheads.

4.1. Member Addition

Addition is a process that is invoked by a member who is authenticated to become a member of a group and subgroup. When a member wants to join a group, perfect backward secrecy must be maintained which means that a newly added user during time T_J should not gain access to any key used to conceal data before T_J. The member addition encompasses three steps, which are; registration, key generation and distribution and key update. In the registration step, a member who wishes to join the group submits a join request along with the individual key to the Access Point, which will pass these details to the Key Server for authentication purpose.

C_n → A: {JOIN_REQUEST} IK

The server checks for the authenticity of the member. Upon confirming the identity of the member, the server generates a new group key. The updated group key is then distributed to all the members including the newly joined member via multicast.

S through A→ C: {JOIN_RESPSONSE} (GK)

4.2. Member Deletion

As mentioned earlier it is rather a complicated process to change the group key when a member is being removed from the group. The member deletion takes effect when the member(s) wants to leave the group, or is forcefully deleted from accessing the future group communications. The basic principle behind the deletion procedure is that the members evicted from or left the group during time T_L should not gain access to any key used to conceal data after T_L, unless they are authenticated to re-join the group. The removal of member(s) is a two step operation, which are: deletion and key update. In the first step, member submits a leave request to the Key Server following which the deletion process takes effect.

C_n → S: {LEAVE_REQUEST}

After the user is deleted from the group the IK of the remaining members and the GK undergo necessary updates as described in 3.4. The only difference in this step would be the changes in the session key of each individual member while their MID remains constant always.

$$\mathbf{S \rightarrow A:}\ \{\text{Key Update Message}\}\ \mathbf{GK} = \sum_{i=1}^{n} \mathbf{IK}$$

$$\mathbf{A \rightarrow C: GK}$$

The key update message from the server contains the list of the member(s) who should be denied any access to the group communication details, unless reauthenticated to join the group.

4.3. Handoff Mechanism

Mobility is the most desirable as well as a convoluted feature of the wireless network. It provides members the anywhere-access to the network. On the other hand, these handoff mechanisms [8] impose various complexities upon the group key management. The logical key structure maintains different KEK's for different area which means that when a member moves from one area to another he needs to know the recent KEK to decrypt the communication contents of the following subgroup. In our approach the members can switch between the access points and still use the same group key, without compromising the security of the group.

5. Discussions

The wireless group communication model must satisfy the following goals:

1. Non-group confidentiality: Members who were not part of the group should not be able to receive the information shared within the group.

2. Key Authenticity: The key server should be assured that it receives requests only from authenticated group members.

5.1. Security of Session Keys

The proposed technique inherits these security features because it never reprocesses a session key during a change in the key tree structure. Another added advantage of our technique is that these session keys which form the integral part of calculating the IK are not based on any long term shared key. Considering that the attacker is successful in generating quite a few session keys and tries to guess the next session key in quick succession, the server can keep track of the total number of erroneously hashed or encrypted messages. When the number of incorrect messages exceeds a predetermined limit the server can then delete the particular member from the group. In order to reactivate the service the member has to start with new set of session keys which mean that the attacker has to start the process of collecting the keys again from beginning in order to generate the individual key. Taking into account the worst scenario, where an attacker has guessed all the right values of session keys and the server has failed to track the fake messages, the short life span of session keys can embark upon this threat in a comprehensively better way. After the time period of current set of session keys is over, the compromised session keys are no longer valid and new set of session keys are generated by the valid parties. So the compromise of session keys does not concern the participating parties in a longer run.

Capturing the values used to generate the session key means that the session key Y_i (the current set of keys) can also be generated. By following our proposed technique the deception of the values involved can still be detected by the server when it receives a HMAC with an old session key.

In the proposed technique it can be noticed that the generated set of session keys do not require long term shared key (MID). Thus even if the long time shared key is compromised it is of no use for generating the individual key as the session key is still unknown. Non-deployment of MID in the proposed technique offers flexibility in making changes to the records without giving much consideration about the network connectivity.

In general, MID is used to identify the member's identity during request for services. Thus, it can be changed into another form by combining it with session key which makes it difficult to be compromised. In our proposed work, member is not required to send MID when requesting a service. With this efficient key generation technique, it is hard to retrieve MID, even supposing an attacker can intercept the message and successfully retrieve the HMAC of Y_i as it is not feasible to run the reverse operation of hash function. In addition, as Y_i is used for two purposes: symmetric encryption and hashing, which makes it hard to retrieve the previous value of Y_i from the intercepted message.

5.2. Security of Individual keys

Each time re-grouping of the members takes place; different IK's are generated based upon new session keys which eventually lead to the change in the group key. Given that IK is formed by performing irreversible hash function on the MID and SK which are never transmitted, it leaves no chance of the key being compromised. The information processed at server end is always encrypted by the session keys before it's transmitted to the member and the session keys used once is never reused ensuring further security of the system. In our proposed work, member is not required to send his MID during the

request for service. The MID is used in combination with the session key only for generating IK which can be done offline.

5.3. Security of Group Keys

With the key tree hierarchy, the generation of a common group key requires only log (T) rounds of key agreement steps between T members. The proposed technique necessitates the generation of session keys (*Yi)*, to carry out the secure communication within the group. These keys are generated on the end terminals of both the member as well as the server. The key Yi is always generated for every session and can be updated when the key structure changes as the result of addition/deletion of group members during a session by the mutual permission of the participating members. During the key distribution phase, each time when a new IK from all the group members is sent over to the key server there is a possibility of an interception. If an attacker successfully computes the GK he would still not be able to communicate with the server unless he has generated the complete set of values denoting Yi. Group key sustained by attacker is of no use if not transmitted with exact value of the session key during information retrieval. Since the group key is computed collaboratively using the individual keys transmitted by all the authenticated members, there is no possibility of the group key being intercepted during any stage of the communication.

6. Conclusion

This paper proposes the use of session keys for secure group communication in which the session key used once, is never reused again. All the members in the secure group are mutually authenticated by executing the key agreement process. A member is not required to send his/her long-time shared key during the request for any service. The generation of individual key which serves as a means of authentication between the server and the member can be done offline. These attributes make the proposed protocol more appropriate for various group communication environments. The proposed work provides very strong security against various attacks based on key compromise in wireless networks; it is very scalable and also balances the performance enhancement of limited resource devices.

7. References

[1] E. Bresson, O. Chevassut, A. Essiari, and D. Pointcheval, "Mutual Authentication and Group Key Agreement for Low-Power Mobile Devices," presented at The Fifth IEEE International Conference on Mobile and Wireless Communications Networks (MWCN '03), Singapore, 2003.

[2] B. DeCleene, L. R. Dondeti, S. Griffin, T. Hardjono, D. Kiwior, J. Kurose, D. Towsley, S. Vasudevan, and C. Zhang, "Secure Group Communications for Wireless Networks," presented at Communications for Network-Centric Operations: Creating the Information Force, IEEE Military Communications Conference (MILCOM), 2001.

[3] T. Hardjono and G. Tsudik, IP Multicast Security: Issues and Directions, Annales de Telecom, July-August 2000, pp. 324-334.

[4] Y. Kim, A. Perrig, and G. Tsudik, "Tree-Based Group Key Agreement," ACM Transactions on Information and System Security (TISSEC), vol. 7, pp. 60-96, 2004.

[5] S. Kungpisdan, B. Srinivasan, and P. D. Le, "Lightweight Mobile Credit-Card Payment Protocol," presented at Proceedings of the 4th International Conference on Cryptology, India, Lecture Notes in Computer Science pp. 295-308,2003.

[6] J. Nam, S. Kim, H. Yang, and D. Won, "Secure Group Communications over Combined Wired/Wireless Networks," presented at IEEE International Conference on Communications, Seoul, 2004.

[7] R. Singh, D. Damodaran and P. D. Le, "Wireless Member-Server Application Model Using Limited Key Generation Technique," To appear in the Proceedings of the Third International Conference on Advances in Mobile Multimedia, Kuala Lumpur, Malaysia, 2005.

[8] Y. Sun, W. Trappe, and K. J. R. Liu, "A scalable multicast key management scheme for wireless networks," IEEE/ACM Transactions on Networking (TON), vol. 12, pp. 653-666, 2004.

[9] B. T. S. Toh, S. Kungpisdan, and P. D. Le "KSL Protocol: Design and Implementation," presented at Proceedings of the IEEE International Conference on Cybernatics and Intelligent Systems, Dec. 2004.

[10] D. Wallner, E. Harder, and R. Agee, "Key Management for Multicast: Issues and Architectures," The Internet Society (ISOC), 1999.

[11] C. K. Wong, M. Gouda, and S. S. Lam, "Secure Group Communications Using Key Graphs," IEEE/ACM Transactions on Networking (TON), vol. 8, pp. 16-30, 2000.

[12] C. Zhang, B. DeCleene, J. Kurose, and D. Towsley, "Comparison of Inter-Area Rekeying Algorithms for Secure Wireless Group Communications," vol. 49, pp. 1-20, 2002.