Scalable Dynamic Key Based Group Key Management System

S.Santhi[1]
Dept. of Computer Science &Engineering,
Anna University
Tiruchirapalli, India

M.Aramudhan[2]
Dept. of Computer Science &Engineering,
Pondicherry University,
Karaikal, India

A.Shanmugasundaram[3]
Dept of Mathematics
VDPC
Nagapattinam, India

Abstract
Multicasting is increasingly used as an efficient communication mechanism for group oriented applications in the internet. It raises a key management problem when data encryption is desired. An efficient key management solution for distributing and changing keys is in great demand for access control of information. In this paper an efficient scalable dynamic key based group key management (SDKGKM) is proposed. SDKGKM has the following advantages. First, it addresses single point failure by introducing panels of controllers. Second, SDKGKM supports scalability by providing subgroup controller panels. Third, it overcomes the drawback of sharing long term secrets by using dynamic keys. Fourth, SDKGKM minimizes the number of keys generated during key generation process and rekeying operation. Fifth, it minimizes the bandwidth cost by adopting an efficient rekeying strategy. A formal analysis of this work is done in this paper. The prototype implementation of this work is done using the java programming language.

Keywords
Group key management, scalability, dynamic key

I. INTRODUCTION

With the increasing ubiquity of the internet and the growing popularity of the IP multicasting, multiparty communication is made a requirement for distributed applications. Group communication applications can use IP multicast to transmit data to all *n* group members using minimum resources. However scalable IP multicast does not provide mechanisms to limit the access to the data being transmitted to authorized group members only. The security challenge for multicast is in providing an effective method for controlling access to the group and its information that is as efficient as the underlying multicast. Many group oriented and distributed applications need security services which includes key management. Such applications need a secure group key to communicate their data. This brings importance to key management techniques. One of the most important issues in multicast security is the group key management (GKM). GKM, which is concerned with the generating and updating secret keys, is one of the fundamental technologies to secure such group communication.

A primary method of limiting access to information is through encryption and selective distribution of the keys used
to encrypt group information. The messages are protected by encryption using the chosen key, which in the context of group communication is called the *group key*. Only those who know the group key are able to recover the original message. Furthermore, the group may require that membership changes cause the group to be rekeyed. Changing the group key prevents a new member from decoding messages exchanged before it joined the group. If a new key is distributed to the group when a new member joins, the new member cannot decipher previous messages even if it has recorded earlier messages encrypted with the old key. Additionally, changing the group key prevents a leaving or expelled group member from accessing the group communication (if it keeps receiving the messages). If the key is changed as soon as a member leaves, that member will not be able to decipher group messages encrypted with the new key.

However, distributing the group key to valid members is a complex problem. Although rekeying a group before the join of a new member is trivial (send the new group key to the old group members encrypted with the old group key), rekeying the group after a member leaves is far more complicated. Therefore, a group key distributor must provide another scalable mechanism to rekey the group.

Efficient group key management protocols should take into consideration the requirements from the points of view of security, quality of service, resources of the keying server, and group member's resources.

1) Security requirements

a) Forward secrecy requires that users who left the group should not have access to any future key

b) Backward secrecy requires that a new user that joins the session should not have access to any old key.

Manuscript received September 5, 2010
Manuscript revised September 20, 2010

c) Collusion freedom requires that any set of fraudulent users should not be able to deduce the current traffic encryption key.

d) Key independence requires that the disclosure of a key should not compromise other keys.

e) Minimal trust requires that the key management scheme should not place trust in a high number of entities.

2) Quality of service requirements

a) Low bandwidth overhead: the rekey of the group should not induce a high number of messages.

b) 1-affects-n: this happens when a single membership change affects all the other members in the group.

c) Minimal delays: many applications are sensitive to the jitters and delays in packet delivery. Hence any key management scheme should minimize delays.

d) Service availability: the failure of a single entity in the key management architecture must not prevent the operation of the whole multicast session.

II. RELATED WORK

A. GKMP [7]

In GKMP [7] the creation and maintenance of a group key is based on centralized group key management approach. In this approach, the key distribution centre (KDC) helped by the first member to join the group creates a group key packet (GKP) that contains a group traffic encryption key (GTEK) and a group key encryption key (GKEK). When a new member wants to join the group, the KDC sends it a copy of the GKP. When a rekey is needed, the group controller (GC) generates a new GKP and encrypts it with the current GKEK .As all members know the GKEK, there is no solution for keeping the forward secrecy when a member leaves the group except to recreate an entirely new group without that member .This is one of the main drawbacks of this GKMP. Another major drawback of this approach is that because it is based on a centralized key management system it suffers from single point failure.

B. Key Management For Multicast – Issues And Architecture[16]

In [16] the authors discuss the various architectural trades-offs involved in the generation, distribution and maintenance of traffic encryption keys for the multicast groups. This paper does not deal with other elements involved in the establishment of a secure connection among multicast participants.

C. Scalable Multicast Key Distribution [4]

In [4] Ballardie proposes a scheme to use the trees built by the core based tree (cut) multicast routing protocol to deliver keys to a multicast group. Any router in the path of a joining member from its location to the primary core can authenticate the member if the router is authenticated with the primary core. There is no solution for forward secrecy other than to recreate an entirely new group without the leaving members.

D. Iolus [12]

In [12] Mittra proposes Iolus, a framework with a hierarchy of agents that splits the large group into small subgroups. A Group Security Agent (GSA) manages each subgroup. The GSAs are also grouped in a top-level group that is managed by a Group Security Controller (see figure 1).

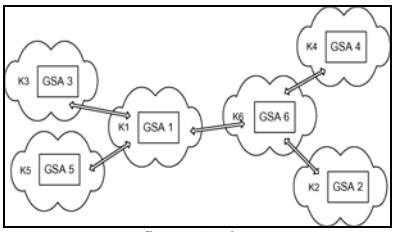

figure 1 Iolus

Iolus uses independent keys for each subgroup and the absence of a general group key means membership changes in a subgroup are treated locally. It means that changes that affect a subgroup are not reflected in other subgroups. In addition, the absence of a central controller contributes to the fault-tolerance of the system. If a subgroup controller (namely GSA) fails, only its subgroup is affected. Although Iolus is scalable, it has the drawback of affecting the data path. This occurs in the sense that there is a need for translating the data that goes from one subgroup, and thereby one key, to another. This becomes even more problematic when it is taken into account that the GSA has to manage the subgroup and perform the translations needed. The GSA may thus become a bottleneck.

E. Kronos[14]

Kronos is an approach driven by periodic rekeys rather than membership changes, which means a new group key is generated after a certain period of time, disregarding whether any member has joined, left or been ejected from the group. Although Kronos does not use a central controller and the subgroup controllers can generate the new keys independently, which makes the system fault-tolerant. It compromises the group security because it

generates the new key based on the previous one. If one key is disclosed, then it compromises all following keys.

F. Group Diffie–Hellman Key Exchange[9]

[9] is an extension for the Diffie-Hellman (DH) key agreement protocol that supports group operations. The DH protocol is used for two parties to agree on a common key. In this protocol, instead of two entities, the group may have *n* members. The group agrees on a pair of primes (*q* and α) and starts calculating in a distributive fashion the intermediate values. The first member calculates the first value and passes it to the next member. Each subsequent member receives the set of intermediary values and raises them using its own secret number generating a new set. Member *n* raises all intermediate values to its secret value and multicasts the whole set. Each group member extracts its respective intermediate value and calculates *k*. The setup time is linear (in terms of *n*) since all members must contribute to generating the group key. Therefore, the size of the message increases as the sequence is reaching the last members and more intermediate values are necessary. With that, the number of exponential operations also increases.

G. Dynamic Keys Based Sensitive Information System[18]

In [18] a dynamic key based secure sensitive information system is proposed. It integrates one time dynamic keys with raw data to protect sensitive information instead of a long term key, and also it uses one time keys to secure communication and authenticate users. The dynamic key based sensitive information system consists of dynamic key generation management (DKGM), authentication & authorization management (AAM) and sensitive information management (SIM). It adopts dynamic key techniques to protect sensitive information in data, verification and communication aspects. DKGM generates dynamic keys for securing communication and raw data. AAM serves as a security shield. It verifies legalization of users by using the dynamic keys generated in DKGM and delegates particular resources for users.SIM acts as a key role. It manages retrieving and assembling sensitive information using generated dynamic keys. [18] Provides strong authentication, secure communication and raw data protection by using dynamic keys to replace long term shared keys. Also fraud deduction and prevention are realized in AAM. DKSIS has the following advantages: Integrating dynamic keys with sensitive data; Enhancing security of communication and authentication; Giving users fine-gained control over their sensitive information; and Providing fraud detection and prevention mechanism.

III. MAJOR SECURITY RELATED SHORTCOMINGS OF THE EXISTING APPROACHES

A. Single Point of Failure

Although the group controller takes the partial role of KDC [7] in generating the desired keys, the role of group controller is being performed by a single node at anytime.

B. Lack of Scalability

Since a single controller is responsible for generating the keys and also validating the entries of the new members, as the group size becomes larger, the node having the role of group controller will be heavily loaded. There is a limit to the number of members a single controller can handle efficiently. There is also no way to prevent a compromised member from being able to permit intruders into the group. The literatures presented do not address this problem.

C. Drawback of Sharing Long Term Secrets

To authenticate individuals and group users in the system the group controller shares a key which is a long term one with the users. This leads to the vulnerability of the key being exposed. [18] Addresses this problem.

D. Generation of a large number of keys.

During key generation and later during rekeying operation a number of keys are generated. As a result the bandwidth cost is increased.

IV. PROPOSED SOLUTION

A survey of the group key management system points out to the fact that no key management system offers an integrated solution to overcome all the drawbacks listed. In the current work each one of the shortcomings is handled in an efficient manner by extending the already existing concepts and at the same time introducing new concepts where possible.

A. Robustness Of The Group Controller

Single point of failure of the group controller as discussed in the literature leads to an unreliable system. Considering the fact that the purpose of the multicast communication is to possibly serve a large number of members, it is of interest to provide a group control mechanism that can survive a single point of failure. Apart from providing reliability of service, the new group-control mechanism should also be able to prevent a compromised group-controller from being able to generate any future keys for the group. Inheriting the idea of Poovendran et al [11] SDKGKM has opted the notion of replacing a group controller with a panel of controllers.

This panel consists of three members at any given time. Among these three, one will serve as the active group controller, with the group keys being generated by two panel members with the constraint that no two panel

members may participate in consecutive key generations. This approach allows every panel member to have only shared key generation authority.

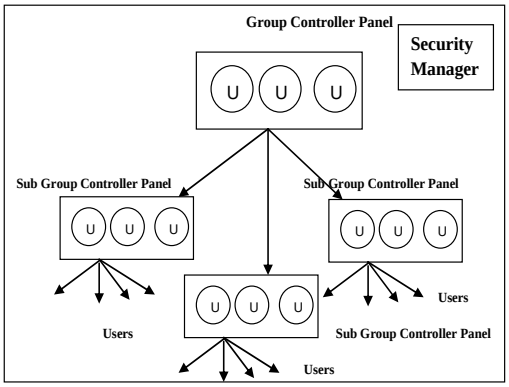

figure 2 Proposed Architecture

Replacing a single group controller with a panel of three members adds more functionality to the panel and reduces the probability of failure of the whole group controller panel. As in the case of the single group-controller, any member can serve as a panel member.

Using the concept of probability it can be proved that the probability of failure of all the three members in the panel at the same time is negligible.

Another theorem namely, the multiplicative theorem of probability when extended to n events assures that there is no loss in the performance of the group controller when it is replaced by a panel of more than one group controller.

B. Scalability

For any multicast group authentication, verification, join authorization, session key parameter negotiation and distribution have to be scalable. These security related operations are independent of group key generation and hence can be allocated to any member of the group other than the group controller. In SDKGKM scalability is achieved by the formation of clusters or subgroups managed by subgroup controller panels [11].
Formation of clusters are based on either of the two criteria namely, role based clustering and temporal clustering. Apart from this a threshold mechanism is also imposed on clustering. All these methods lead to small subgroups or clusters.

C. Minimal Key generation

In general the rekeying of a dynamic group generates heavy network traffic. This leads to bandwidth overhead as the group size increases. One of the main objectives of any multicast group is to minimize the bandwidth requirements. In [10] the authors propose STauth, a secure, scalable and efficient key management protocol for location based services. Using the idea of STauth, SDKGKM deals with the formation of clusters based on temporal authorization. The subgroup controller panel managing each cluster rekeys the subgroup. As a result the number of keys generated is less and the scalability is increased.

D. Rekeying cost reduction

In general the group controller uses the communication, computation and storage resources for distributing the session key to the group of n members. SDKGKM aims to reduce the rekeying cost. This approach extends the secure lock method [15] of rekeying. The rekeying is performed by the subgroup controller. Because of the small size of the subgroups the computation overhead is also reduced.

E. Dynamic Key Based Approach

When a user joins a group, for backward secrecy, a new group key is generated, encrypted by a shared unique key and sent to the user. In order to prevent the group key from risks associated with the compromise of long term unique shared cryptographic keys, dynamic keys are used to overcome the threats.

A dynamic key [17] is a single-use symmetric key used for generating tokens and encrypting messages in one communication flow. There are three primary reasons [17] for using dynamic keys used in SDKGKM. Firstly, the use of long term share keys makes sensitive information systems vulnerable for adversaries. However, using dynamic keys makes attacks more difficult. Secondly, most sound encryption algorithms require cryptographic keys to be distributed securely before enciphering takes place. However, key distribution is one of the drawbacks of symmetric key algorithms. Although asymmetric key algorithms do not require key distribution, they are slow and susceptible to brute force key search attack. Therefore, the use of asymmetric key algorithms to distribute an encrypted secret for another is only once. Then dynamic keys are generated based on the secret and other key materials. It can improve the overall security considerably. Last but not least, security token can be generated by either long term symmetric keys or nonce dynamic keys. Even though both methods generate variational tokens every time, dynamic key method is more difficult to break than long term key method.

In SDKGKM, dynamic keys eliminate shared long term unique keys between group members and key controllers.

V. CONCLUSION

In this paper a robust, scalable, dynamic key based extension to the Group Key Management protocol for multicast communication has been proposed. By replacing the single group controller by a panel of controllers, the threat of single node failure is eliminated.

This scheme also helps in the removal of a compromised panel member. Introduction of the sub-group panels help in realizing the scalability of the system**.** An efficient rekeying strategy reduces network traffic and reduces bandwidth cost. A novel clustering strategy minimizes the number of keys generated. The use of Dynamic keys ensures the security of the information in the system. The prototype system is implemented in java.

Future directions of this work include:

1. Extend the scheme to support spatial constraints.
2. Extend the scheme to support non-access hierarchies.

REFERENCES

[1] M. J. Atallah, M. Blanton, and K. B. Frikken, "Incorporating temporal capabilities in existing key management schemes," in Proc. ESORICS, 2007, pp. 515–530.

[2] M. J. Atallah, M. Blanton, and K. B. Frikken, "Efficient techniques for realizing geo-spatial access control," in Proc. Asia CCS, 2007, pp.82–92.

[3] M. Atallah, K. Frikken, and M. Blanton, "Dynamic and efficient key management for access hierarchies," in Proc. ACM CCS, 2005, pp.190–202.

[4] A. Ballardie, University College London "Scalable Multicast Key Distribution" Request for Comments: 1949

[5] Chung Kei Wong, Mohamed Gouda, Simon S. Lam "Secure Group Communications Using Key Graphs." Appeared in the proceedings of ACM SIGCOMM'98

[6] David A. McGrew and Alan. T. Sherman,"Key Establishment in Large Dynamic Groups Using One-Way Function Trees". Research work funded by DARPA.

[7] H. Harney, C. Muckenhirn– "Group Key Management Protocol (GKMP) Architecture." Request for Comments: 2094

[8] H. Harney, C. Muckenhirn "Group Key Management Protocol (GKMP) Specification." Request for Comments: 2093

[9] Michael Steiner, Gense Tsudik, Michael Waidner, "Diffie-Hellman Key Distribution Extended to Group Communication"3rd ACM Conference on Computer and Communication Security 1996

[10] Mudhakar Srivastsa, Arun Iyengar, Jian Yin and Lig Liu "Scalable Key Management Algorithms for Location – Based Services". IEEE / ACM Transcations on Networking Vol. 17 No.5 October 2009.

[11] R. Poovendran, S. Ahmed, S. Corson, J. Baras "A Scalable Extension of Group Key Management Protocol". Prepared under the US Army Research Laboratary Sponsorship.

[12] Suvo Mittra "Iolous : A Framework for Scalable Secure Multicasting" Proceedings of ACM SIGCOMM '97. September 1977 France.

[13] Sandro Rafaeli, Lancaster University "A Decentralised Architecture for Group Key Management". Technical Report, September 2000

[14] Sandro Rafaeli and David Hutchison "A survey of Key Management for Secure Group Communication". ACM Computing Surveys Vol.35, No.3, September 2003 PP 309 – 329

[15] R. Srinivasan, V. Vaidehi, R. Rajaraman, S. Kanagaraj, R. Chidambaram Kalimuthu and R. Dharmaraj "Secure Group Key Management Scheme for Multicast Networks" – International Journal of Network Security, Vol. 11, No.1 PP 30 – 34, July 2010.

[16] D.Wallner, E. Harder, R. Agee, "Key Management for Multicast : Issues and Architecures". Request for Comments: 2627

[17] Xianping Wu , Huy Hoang Ngo ,Phu Dung Le and Balasubramaniam "Novel Authentication & Authorization Management for Sensitive Information Privacy Protection using Dynamic Key Based Group Key Management" International Journal of Computer Science and Application Vol. 6, No.3 PP 57 – 74 2009

[18] Xianping Wu, Phu Dung Le and Balasubramaniam Srinivasan, School of Information Technology, Monash University, Melbourne, Australia."Dynamic Keys Based Sensitive Information System". 9th International IEEE Conference for Young Computer Scientists.