Multicast Transmission Key Management Approachon

Anirudha Kumar Pandey [#1], Dr. Sipi Dubey[#2]

[#] *Deptt. Of Computer Science & Engineering, RCET, Bhilai Chhattisgarh, Bhilai - India*

[1] ani.kumar.pandey@gmail.com

[2]drsipidubey@gmail.com

Abstract- Multicast networking support is becoming an increasingly important future technology area for both commercial and military distributed and group-based applications. Integrating a multicast security solution involves numerous engineering tradeoffs. Various techniques that have been proposed to support multicast security are discussed and their relative merits are explored.In this paper we are proposing the multicast security based on key management.

Key Word: Multicast, Datagram, Unicast, UDP.

I. INTRODUCTION

The future of military communications is rapidly converging on visions of connectivity such as the Global Information Grid (GIG) and Joint Tactical Radio System (JTRS). In these visions, combinations of wire line and wireless communications are used to provide many to many collaborative communications between troops and sensors in theater, and command and control systems located thousands of miles away. Large numbers of entities participating in these communications warrant using efficient protocols such as multicast in order to reduce network congestion. When members of a multicast group need to receive the same information securely and are allowed to dynamically join or leave the group, security entails not only distribution of a secret among many but may also be concerned with confidentiality of information as the membership changes. Military communications require different levels of security based on policies governing the shared information. When considering strict secrecy policies, it is important that a new member to the multicast group not be able to decode previous information that was transmitted (backward confidentiality) and a current member who leaves (or is ejected) not be able to decode future information that will be transmitted (forward confidentiality). As the level of confidentiality is relaxed, the amount of forward and backward confidentiality is also eased. Secure group communications also seek to prevent collusion, in which a set of members exchange information to gain additional unauthorized access. Another feature important to secure groups is containment; compromise of one member should not compromise the entire group. Approaches that work in unicast transmissions, such as SSLand VPNs do not extend to a multicast group. VPNs do support multicast but only by unicasting the data to each wireless VPN client, effectively removing the bandwidth efficiency of multicast. The problem of secure group communication has been the subject of much recent research with both an IRTF research group, the Group Security Research Group (GSEC) and an IETF Working Group, Multicast Security Working Group(MSEC) addressing the issue. Most of the work in these groups has been directed toward wired networks but the issues therein identified also apply to wireless networks. Features that have been identified as necessities of a key management system for secure mobile multicast groups include scalability, data confidentiality, data integrity, source authentication, forward and backward confidentiality, collusion resistance, and compromise resistance. While the following features enhance the performance of any group security scheme, they are particularly important to compensate for the constraints of mobile wireless networks: minimal messaging bandwidth usage, minimal security related computation, storage efficiency of keys, and low latency for rekey messages. [WCetal] classifies secure multicast protocols into three categories: centralized flat schemes, distributed flat schemes, and hierarchical schemes. [DMS] considers the security and scalability issues of each category with the following analysis. Centralized flat schemes do not scale since one change affects all members, known as the ‘1 affects n’ scalability problem. Distributed flat schemes are vulnerable to collusion attacks. Hierarchical schemes using a hierarchy of keys also suffer from the ‘1 affects n’ scalability problem. However, protocols with a hierarchy of nodes responsible for key distribution, but not data distribution, address the scalability and security risks of the other schemes. Applications such as conferencing, distributed interactive simulations, networked gaming, and news dissemination are group oriented. In these applications, it is necessary to secure the group communication as the data are sensitive or it requires the users to pay for it. In the algorithms for secure group communication, a group key is shared by all the users. The group key is used to encrypt data transmitted to the group. The group Membership is dynamic. When group membership changes, to protect the confidentiality

of the current users, a new group key needs to be shared by the users. The dynamics of the group membership can be handled under two settings. In the first setting, a central group controller manages the group membership and the users do not have the necessity to communicate among themselves. Scenarios like pay TV, news dissemination, stock information, etc., are in this category. In these scenarios, typically, the group size is large and geographically disparate. In the second setting, the group members collaborate to agree upon a common group key [1]. Applications like conferencing and distributed interactive simulation fall under this category. The group sizes in such applications are typically small and justify the usage of the relatively high end computation required by the group key agreement techniques [2]. In this work, we consider the first setting where a large group of users is managed by a group controller and consider the cost of membership handling in such applications. When a user is admitted to the group, the group controller changes the group key and securely uncast it to the joining user. To send the new group key to the current users, the group controller encrypts it with the old group key and multicasts it to them. Thus, the cost of rekeying for the group controller, due to a joining user is small. However, when a user is revoked, i.e., the user leaves or is forcefully removed from the group, the group controller needs to securely unicast the new group key to each of the remaining users. Toward this, the group controller encrypts the new group key with the personal keys of each of the remaining users and unicasts each message to the respective user. The cost of this process is O(N) symmetric key encryptions and O(N) messages. Thus, for a large group, revoking users from the secure group is an expensive operation.
Many solutions have been proposed for efficiently handling a single membership change, i.e., a single join or revocation of a user. In these solutions, for a group of N users, the group controller distributes the new group key in O (logN) encrypted messages. We note that in these solutions, the rekeying cost, i.e., number of encryptions performed and messages transmitted by the group controller, for a joining user is increased from two to O(logN). However, techniques suggested reduce the join cost to nearly constant and as such have been used by other approaches [5], [6]. On the other hand, the cost for revoking a user is reduced from O(N) to O(logN) encrypted messages. However, to handle multiple membership changes, the group controller repeats the process of revocation for each revoked user. Optimizations such as batch or periodic rekeying reduce this cost to some extent. However, even in these solutions, the cost of revocation is high. Moreover, as the group controller needs to interrupt the group communication during the rekeying, the resulting delay can be unreasonable for many applications. Thus, efficient distribution of the new group key for multiple membership changes is a critical problem in secure group communication.

One approach to revoke multiple users is to associate a key with every nonempty subset of users in the group. Thus, if one or more users are revoked, the group controller uses the key associated with the subset of the remaining users to encrypt the new group key and transmits the new group key to them. The advantage of this approach is that the Communication overhead is only one message for revoking any number of users. However, the number of keys stored by the group controller and the users is exponential in the size of the group. In this paper, we describe a family of key management algorithms that reduce the cost due to multiple user revocation while keeping the storage cost manageable. The goal of the paper is to evaluate trade-off between storage and revocation cost. Storage is computed in terms of keys that each user maintains [9]. And revocation cost is computed in terms of the encryptions performed, and the number of messages transmitted, by the group controller. Similar to the algorithms, we assume that the communication from the group controller is broadcast in nature. Using our algorithms, the group controller can efficiently distribute the group key.

II. RELATED WORK

Other approaches to address the problem of revoking multiple users are proposed in [3]. The group controller maintains a logical hierarchy of keys that are shared by different subsets of the users. To revoke multiple users, the group controller aggregates the entire necessary key updates to be performed and processes them in a single step. The group controller interrupts the group communication until all the necessary key updates are performed, and then, distributes the new group key to restore group communication. This interruption to group communication is undesirable for real-time and multimedia applications. To handle multiple group membership changes, the group controller performs periodic rekeying, i.e., instead of rekeying whenever group membership changes, the group controller perform rekeying only at the end of selected time intervals. However, the revoked users can access group communication until the group is rekeyed.
This can either cause monetary loss to the service provider or compromise confidentiality of other users. The group controller maintains a logical hierarchy of keys similar to the solution. To revoke multiple users, the group controller distributes the new group key by using keys that are not known to the revoked users. However, this solution achieves a good rekeying cost only if the size of the revoked users is either very small or very large. In the above schemes, the logical key tree structure tends to become unbalanced after some membership changes and results in tree which has large height (O (N)). As the height of the tree determines the rekeying cost, several approaches [7] have been proposed to address this issue. These approaches focus on algorithms for reorganizing the tree structure that becomes unbalanced after a few membership changes. However, the basic

rekeying algorithm. The approaches in these works are orthogonal to our algorithms in that the approaches from these works can be used to balance the tree used in our algorithms. The authors describe an information theoretic approach for analyzing key-tree based protocols and show interesting relationships among the storage cost, the number of rekeying messages, and the resistance against colluding users. They describe an optimal key distribution protocol which is weakly collusion resistant, i.e., it cannot tolerate collusion of two users.

III. KEY MANAGEMENT ROLE

Key management plays an important role enforcing access control on the group key (and consequently on the group communication). It supports the establishment and maintenance of key relationships between valid parties according to a security policy being enforced on the group [McDanielet al. 1999]. It encompasses techniques and procedures that can carry out:

—*Providing member identification and authentication*. Authentication is important in order to prevent an intruder from impersonating a legitimate group member. In addition, it is important to prevent attackers from impersonating key managers. Thus, authentication mechanisms must be used to allow an entity to verify whether another entity is really what it claims to be.

—*Access control*. After a party has been identified, its join operation should be validated. Access control is performed in order to validate group members before giving them access to groupcommunication1 (the group key, in particular).

—*Generation, distribution and installation of key material*. It is necessary to change the key at regular intervals to safeguard its secrecy [Schneier 1996]. Additional care must be taken when choosing a new key to guarantee key independence. Each key must be completely independent from any previous used and future keys, otherwise compromised keys may reveal other keys. The key secrecy can be extended to membership changes. When a group requires backward and forward secrecy [Kim et al. 2000], the key must be changed for every membership change. Backward secrecy is used to prevent a new member from decoding messages exchanged before it joined the group. If a new key is distributed for the group when a new member joins, it is not able to decipher previous messages even if it has recorded earlier messages encrypted with the old key. Forward secrecy is used to prevent a leaving or expelled group member to continue accessing the group's 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. As multicast is being used for group transmission, it is generally assumed that multicast should also be used to rekey the group.[2] It is not reasonable to consider transmitting data using a scalable multicast communication and rekeying the members under a non-scalable peer to- peer communication. If the group has thousands of members, sending them a new key one by one would not be efficient. Although rekeying a group before the join of a new member is trivial,[3] rekeying the group after a member leaves it is far more complicated. The old key cannot be used to distribute a new one, because the leaving member knows the old key. A group key distributor must therefore provide other mechanisms to rekey the group using multicast messages while maintaining the highest level of security possible.

IV. KEY MANAGEMENT ALGORITHM

4.1 The Basic Structure

We arrange a group of K users as children of a rooted tree, as shown in Fig. 1a. Let R be the root node. We use the tuple <R, u1, u2, ...,uK> to denote the basic structure. The key management algorithm we use for the basic structure is the complete key graph algorithm. In this algorithm, for every nonempty subset of users, the group controller provides a unique shared key which is known only to the users in the subset. The group controller gives these keys to the users at the time of joining the group. Of the keys that auser, say ui, receives: 1) one key is associated with the set { u1, u2, ...,uK}, and hence, is known to all the users and 2) one key is associated with the set {ui} The former key, say kR, is the group key, whereas the latter key is the personal key. Thus, the number of keys stored by the group controller is 2K-1 and the number of keys held by each user is 2K-1. Now, we consider the process of rekeying in this scheme when one or more users are revoked from the group. The proof of the following theorem describes the simple rekeying process for user revocation:

3.2The Hierarchical Key ManagementAlgorithm

In our hierarchical algorithm, we compose smaller basic structures in a hierarchical fashion. To illustrate the hierarchical structure, consider the sample structure <R, R1, R2. Rd> shown in Fig. 1b, where each further consists of the basic structure <Ri, ui1, ui2, . , uid>. The parameter d is the number of elements in a basic structure and can be considered as the degree of the hierarchy. We note that the degree can be different for different nodes in the hierarchy. However, for the sake of simplicity, in this section, we assume that the nodes in the hierarchical structures have a uniform degree d.

Now, each of the basic structures of the form <Ri, ui1, ui2. . . uid > is associated with the shared keys. The structure at next higher level, <R, R1, R2. Rd>, is also associated with shared keys. The personal key associated with Ri, 1 < i < d in structure <R, R1, R2, . . ., Rd> is the same as the group key of the structure <Ri, ui1, ui2, . . . uid>. Furthermore, the structure <R,R1,R2 . . .Rd> is associated with shared keys. Now, each user in the basic structure <R, ui1, ui2 . . . uid > is provided with any shared key that is provided to Ri in the structure <R, R1,R2, . . .Rd>. To illustrate our hierarchical algorithm, we consider

four examples for d = N, 2, 3, 4. In the hierarchical structure, we denote the key associated with asubset <a, b . . .z> by Kab…Z .

V. PROPOSED APPROACH

Step 1: Start the simulation.
Step2: By Default we are forming three groups G1, G2 & G3.
Step3: G1 Consists 5 Nodes, G2 Consists 5 Nodes & G3 Consists 5 Nodes.
Step4: A node want to join any group must follow these following rules

1. Send a request to join any group as JOIN REQ.Groups send the REPLY REQ to particular node with unique key.
2. Node forwards this key to particular group for aunthication. Group maintains the set of aunthicate keys for aunthication purpose.
3. When the key match with than node join the group for communication

Step5: A node want to leave the group must follow these following rules

1. Group must ensure that leaving node dosent have any on going transmission.
2. If transmission is going on than first complete this transmission than leave the group.
3. It these two conditions are true than node can leave the group.

Step6: End the Simulation.

VI CONCLUSION & FUTURE WORK

In this paper, we presented a method for designing the multicast key management tree for the mobile wireless environment. By matching the key management tree to the cellular network topology, a reduction in communication burden of the rekeying messages was observed compared to trees that are independent. We have developed a key management system for secure multicast group communications in mobile network environments. A demonstration platform is implemented of the topology. The flexibility of our system allows it to be more efficient, scaleable, and secure than alternatives. Future areas of research include incorporation of non-repudiation (e.g., through the use of digital signatures) in the system.

.REFERENCES

[1] Y. Kim, A. Perrig and G. Tsudik, "Tree- Based Group Key Agreement," ACM Trans. Information and System Security, vol. 7, no.1, pp.60-96, 2004.
[2] M. Manulis, "Security-Focused Survey on Group Key Exchange Protocols," Report 2006/395, Cryptology ePrint Archive, http:// eprint.iacr.org/, 2006.
[3] F. Zhu, A. Chan, and G. Noubir, "Optimal Tree Structure for Key Management of Simultaneous Join/Leave in Secure Multicast," Proc. Military Comm. Conf. (MILCOM), 2003.
[4] W.H.D. Ng, M. Howarth, Z. Sun, and H. Cruickshank,"Dynamic Balanced Key Tree Management for Secure Multicast Communications," IEEE Trans. Computers, vol. 56, no. 5, pp. 577-589, May 2007.
[5] S. Zhu, S. Setia, S. Xu, and S. Jajodia, "Gkmpan: An Efficient Group Rekeying Scheme for Secure Multicast in Ad-Hoc Networks," Proc. IEEE Mobiquitos '04, pp. 42-51, 2004.
[6] Y. Sun, W. Trappe, and K.J.R. Liu, "A Scalable Multicast Key Management Scheme for Heterogeneous Wireless Networks," IEEE/ACM Trans. Networking, vol. 12, no. 4, pp. 653-666, Aug. 2004.
[7] M.H. Heydari, L. Morales, and I.H. Sudborough, "Efficient Algorithms for Batch Re-Keying Operations inSecure Multicast," Proc. 39th Ann. Hawaii Int'l Conf. System Sciences, vol. 9, 2006.
[8] J.H. Cheon, N. Jho, M. Kim, and E. Yoo, "Skipping,Cascade, and Combined Chain Schemes for Broadcast Encryption," IEEE Trans. Information Theory, vol. 54, no.11, pp. 5155-5171, Nov. 2008.
[9] Bezawada Bruhadeshwar and Sandeep S. Kulkarni,"Balancing Revocation And Storage Trade-Offs In Secure Group Communication" IEEE Transactions on Dependableand Secure Computing, Vol. 8, no. 1, Jan-Feb. 2011.
[10] C. Wong, M. Gouda, and S. Lam, "Secure group communications using
key graphs," IEEE/ACM Trans. on Networking, vol. 8, pp. 16–30, Feb.2000.
[11] R. Canetti, J. Garay, G. Itkis, D. Miccianancio, M., and B. Pinkas, "Multicastsecurity: a taxonomy and some efficient constructions," in INFOCOM'99, 1999.
[12] W. Trappe, Jie Song, R. Poovendran, and K.J.R. Liu, "Key distribution for secure multimedia multicasts via data embedding," in ICASSP'01, May2001.
[13] K. Brown and S. Singh, "Relm: Reliable multicast for mobile networks, "Computer Communication, vol. 2.1, no. 16, pp. 1379–1400, June 1996.
[14] D. Balenson, D. McGrew, and A. Sherman, "Key management for largedynamic groups: one-way function trees and amortized initialization," InternetDraft Report.
[15] M. Rajaratnam and F. Takawira, "Nonclassical traffic modeling and performance analysis of cellular mobile networks with and without channel reservation," IEEE Trans. on Vehicular Technology, vol. 49, no. 3, pp. 817–834, 2000.
[16] M. M. Zonoozi and P. Dassanayake, "User mobility modeling and characterization of mobility patterns," IEEE Journal on Selected Areas in Communications,vol. 15, no. 7, pp. 1239–1252, 1997.
[17] D. Hong and S. S. Rappaport, "Traffic model and performance analysis for cellular mobile radio telephone systems with prioritized and nonprioritizedhandoff procedures," IEEE Trans. on Vehicular Technology, vol. VT-35,
no. 3, pp. 77–92, 1986.