http://ansinet.com/itj

ITJ

ISSN 1812-5638

INFORMATION TECHNOLOGY JOURNAL

ANSInet
Asian Network for Scientific Information
308 Lasani Town, Sargodha Road, Faisalabad - Pakistan

Information Technology Journal 6 (2): 247-250, 2007
ISSN 1812-5638

Effective Rekeying Architecture for Dynamic Multicast Group

[1]V. Vijayaraghavan and [2]R.S.D. Wahidabanu
[1]Department of CSE, Sona College of Technology, Salem-636 005, India
[2]Department of CSE, Govt. College of Engineering, Salem-636 011, India

Abstract: Developments in the area of group communication have been enormous. Multicasting is the group communication paradigm of future, as it saves bandwidth considerably. The main features that makes multicasting as an attractive one, also makes security and the related tasks still a subject of research. Membership in a multicast group is highly dynamic, with receivers entering and leaving the multicast session without explicit knowledge of other hosts. This study concentrates on secure data transmission among highly dynamic multicast groups. In dynamic groups, key management is a complicated procedure since users join and leave at any time. In the proposed architecture to reduce the number of key updates whenever the member joins or leaves, intermediate node in the tree maintains two keys. One is to commune with its parent node and another is to its children. The results show the performance of this architecture outstrips the existing one in the aspects of rekeying overhead.

Key words: Group communication, multicast security, key management, rekeying, Unicast

INTRODUCTION

Secure group communication is an increasingly popular research area, which has been receiving much attention in recent years. As a rapid growth of the internet, the need for simultaneous delivery of data to multiple recipients also increases. The scalable solution for this is multicast communication. Multicast (Judge and Ammar, 2003; Yacine *et al*., 2005) is a set of technologies which is used to deliver data to multiple recipients in an efficient manner. The lightweight join model, which is advantageous for many applications, also influences several security issues. As the group membership is open, any interested host can join the multicast session without any interference which provides many possibilities for eavesdropping. Furthermore, any malicious host can occupy the resources by joining the group, which may degrade the performance. Group key management and multicast receiver access control (Judge and Ammar, 2002a) are the solutions proposed to guard against these attacks. Another serious problem is multicast model can transfer any amount of data delivered to the multicast address to the entire group. This indicates that any host can send data to the multicast group which leads to the following vulnerabilities. The group members must verify that the message is from intended source. This functionality is very well proposed by multicast source authentication (Perring *et al*., 2001) solutions and the next issue is unauthorized data must be eventually avoided from being delivered to the multicast group members. Thus denial of service attacks may be restricted by implementing multicast sender access control. The directive and beneficiary feature of multicast is that all members receive all packets send to the group which makes no individualization of the received data. Sometimes individualization can be used to provide security with the help of fingerprinting. Fingerprinting (Judge and Ammar, 2000b) is the technique in which receiver information is embedded along with the data to prevent unauthorized duplication and propagation of data. However fingerprinting used in unicast environment do not efficiently work in multicast environment because multiple user share the same identity. So the proposed multicast fingerprinting techniques aim at providing a unique fingerprinting in a multicast environment which ensures the efficiency of the technique.

The focus of this study is to propose a viable key management protocol for highly dynamic groups. The proposed protocol considers the following constrains, work well in both small and large groups, as well as static and dynamic groups. Achieve better efficiency compared to the existing protocols, in terms of the number of rekey operations while joining to or leaving from the group. Reduce overhead on the central server of the group and distribute key computing responsibilities across the group. Compute the new keys and transmit them only when ever required. It reduces the number of times a new key is generated such that the network is not overloaded with keys.

Corresponding Author: V. Vijayaraghavan, Department of CSE, Sona College of Technology, Salem-636 005, India
Tel: 919894549259

KEY MANAGEMENT

The notion of key management is secure distribution of secret key between legitimate users. Because of the unique property of multicast the same solution applied for unicast cannot be applied for multicast. The important property of key management is dynamics and scalability. Key has to be updated whenever any change in the membership to assure forward and backward secrecy. Once the member has left the group, probability of revealing the key is high. To ensure secrecy, group key needs to be updated. Also single change in membership should not affect the entire group, which assures scalability. There are several architectures existing for key management. Each one serves for different applications. In centralized (Blundo *et al*., 1998; Harney and Muckenhirn, 1997) architecture, a single server generates the secret key and distributes them to all the members of the group, but it fails to address the issue availability. As it depends on single node, protocol can not be used if the node fails. The problem of distributed (Caronni *et al*., 1998) architecture is to trust all the nodes. Trusting all the members is vulnerable to security attacks inside the group. In dynamic groups, providing scalability is very difficult, this also increases computational overhead whenever there is a change in group. The goals of the key management (Steiner *et al*., 1998) are to ensure that the knowledge of group key is restricted to group members only and secret keys are updated regularly to avoid key compromises from prior session keys. Lastly reduces the utilization of resources, like storage overhead and bandwidth utilization. To achieve these goals the proposed architecture uses tree based key hierarchy architecture (Wallner *et al*., 1999; Wong *et al*., 1997).

PROPOSED ARCHITECTURE

As the members join and leave the group at their will without any explicit information to other group members, the process of key management become very complex. In order to maintain forward and backward secrecy (group members are only able to read the content) secret key has to be updated whenever any change in the group, which leads to significant computational overhead. To reduce that, proposed subgroup approach uses two keys by intermediate nodes as it participates in two subgroups. One key is used to commune with its parent and another key is to commune with its children. Key sharing problem is eliminated by using Diffie-Hellman key agreement protocol. Every subgroup member distributes their share to generate the secret keys using this protocol. If there is any change in the group, it is enough to change the secret key of few subgroups.

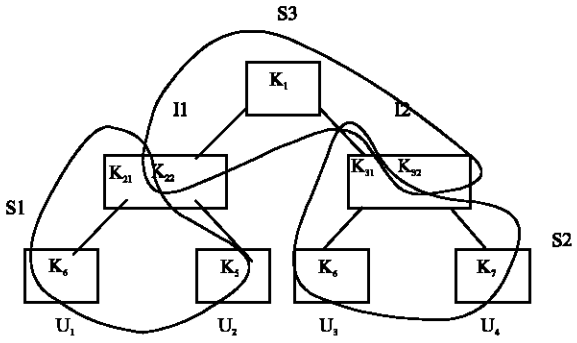

Fig. 1: Subgrouping key structure

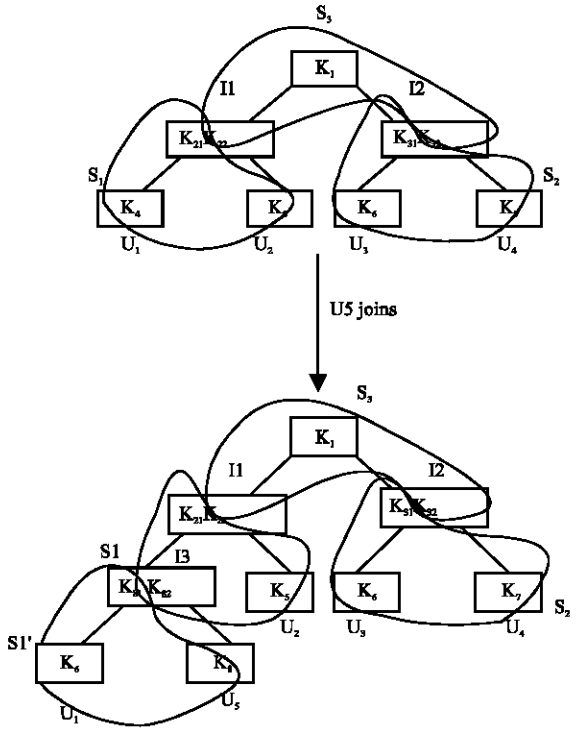

Fig. 2: Key structure for member join event

In this proposed architecture tree structure is used for key distribution. Each node in the tree represents a secret key share and the members are associated with the leaf nodes. To reduce the rekeying update, subgrouping approach is used. Figure 1 shows U_1, U_2 and I_1 form a subgroup S_1, similarly two more subgroups S_2, S_3 are formed. Each node will contribute their share to generate the secret for that subgroup. Diffie-Hellman protocol is used to compute the secret key. For the subgroup S_1 using the contribution of k_4, k_5 and k_{21} secret key of the subgroup is generated. For subgroup S_3 shares k_{22}, k_{32} and k_1 are used. I_1 has two keys k, k^1 one is commune with root and another is used to commune with its leaf node.

The advantage is if any change in the membership, it is enough to change secret key of few subgroups only.

Key structure for member join event: If any new member joins the group, it sends the request to the server. The server finds a proper place for new member in the tree. If the insertion point is the right most node, only that particular subtree's secret key needs to be updated. Otherwise if the tree is balanced the new member creates intermediate node and join as its leaf node. Figure 2 shows new member U_5 wants to join in the tree, it creates intermediate node I_3 and new subgroup is formed. Now secret key of subgroup S_1 need to be updated using the shares k_{21}, k_{81} and k_5, secret key of the subgroup S_1^1 has to be generated using the shares k_{82}, k_4 and k_8 and rest of the tree remains unaltered.

Key structure for member leave event: It is the simplest process when compared to member joins. The user contacts the server when he decided to leave the group. The server may remove the existing parent node of the user who decided to leave and deletes his leaf node. Then it generates the rekey message and updates its parent. In Fig. 3 the member U_4 wants to leave. Therefore, the node U_4 and its parent are removed from the tree. The Sibling node U_3 is attached to the root. Now the secret key of S_3^1 is updated using the shares k_1, k_{22} and k_6.

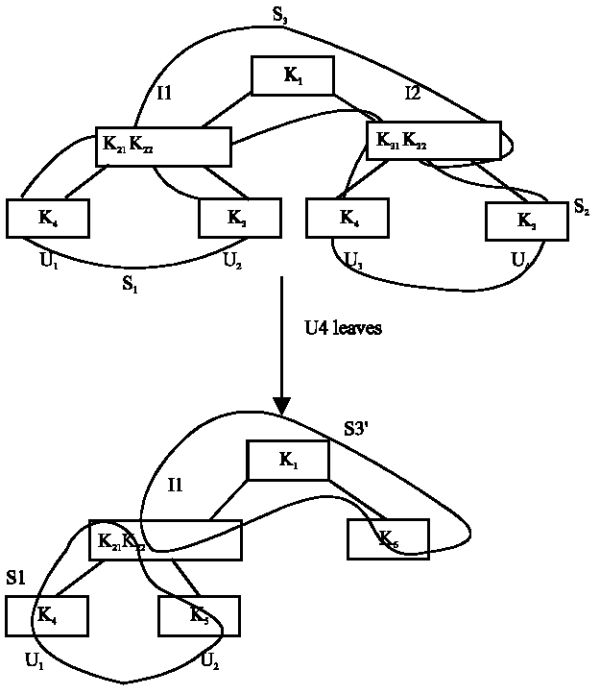

Fig. 3: Key structure for member leave event

PERFORMANCE ANALYSIS

Table 1 shows comparison of different key management schemes. It indicates that the proposed protocol combines the advantage of both the methods

Table 1: Comparison of different key management schemes

	Hierarchical	Iolus	Proposed architecture
Tree based	Yes	No	Yes
Forward secrecy	Yes	Yes	Yes
Backward secrecy	Yes	Yes	Yes
Scalability	Yes	Yes	Yes
Key-Symmetric/ Asymmetric	Asymmetric	Asymmetric	Symmetric
Number of rekey messages received by each member join/leave	O(log n)	O(1)	O(1) only by its parent and sibling
Number of keys stored in each member	O(log n)	O(1)	O(2) for intermediate nodes O(1) for other nodes

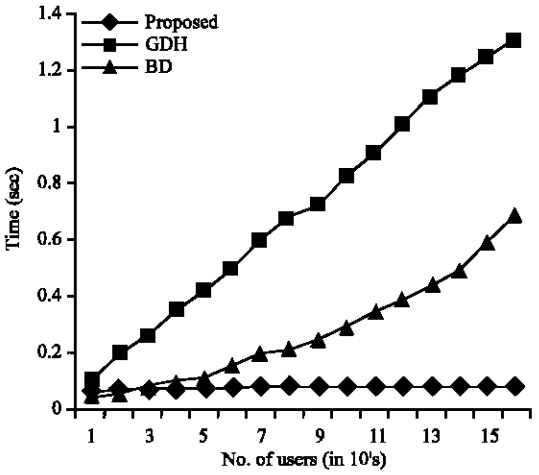

Fig. 4: Join delay comparison

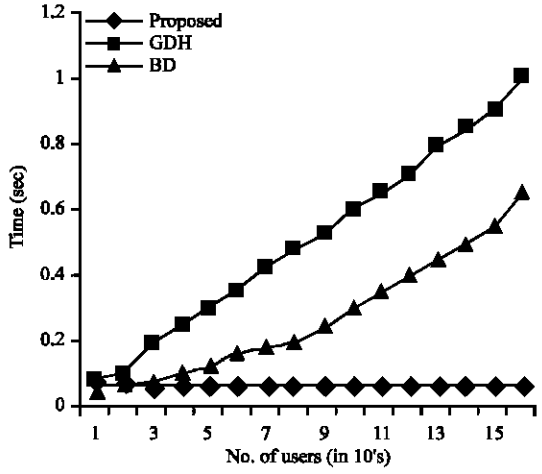

Fig. 5: Leave delay comparison

hierarchical and Iolus (Mittra 1997). As it uses symmetric key management the encryption overhead is also considerably reduced, where as the key sharing problem is eliminated by Diffie-Hellman protocol (Schneier 1996). Also the table shows that the existing method has less rekeying overhead. The analysis of the communication cost in terms of delay between the proposed protocol and other group key agreement protocols including Group Diffie- Hellman (Steiner *et al*., 2000) (GDH) and Brumester-Desmedt (BD) (Burmester and Desmedt, 1998) is made. The results are shown in the graph for join and leave events. Delay is calculated as the time required by the node to join the group and to compute the secret key.

Join operation: Figure 4 shows the comparison for the member join operation for different protocols. In our protocol as the key updation need not be propagated to the entire tree, it has less delay. As the Group Diffie-Hellman involves many modular exponentiations, the delay is very high. Brumester-Desmedt also has more delay, as the hidden cost is of major concern

Leave operation: Figure 5 shows the delay required by different protocols for leave operation. As BD needs to restart whenever, there is a change in membership, delay is significantly high. GDH also involve considerable delay. But in the proposed method delay is very minimal as it needs to update only few subgroups.

CONCLUSIONS

The proposed key management protocol is secure, scalable and efficient. Subgrouping approach is used to reduce the key updation overhead. By using two keys at the intermediate nodes, key updation is restricted only to the specific subgroups where the membership change occurred. This multicast key management protocol has less computational and communication cost of group rekeying than the previous schemes and also it is effective for large and highly dynamic multicast group.

REFERENCES

Blundo, C., A.D. Santis, A. Herzberg, S, Kutten, U. Vaccaro and M. Yung, 1998. Perfectly-secure key distribution for dynamic conferences, Information and Computation, 146:1-23.

Schneier, B., 1996. Applied Cryptography, 2nd Edn., John Wiley and Sons, Inc.

Burmester, M. and Y. Desmedt, 1998. A Secure and Efficient Conference Key Distribution System, In Advances in Cryptology, A.D. Santis, (Ed.), EUROCRYPT '94, number 950 in lecture notes in computer science, International Association for Cryptologic Research, Springer-Verlag, Berlin, pp: 275-286.

Caronni, G., M. Waldvogel, D. Sun and B. Plattner, 1998. Efficient Security for Large and Dynamic Groups, Technical Report TIK Technical report No 41,Computer Engineering and Networks Laboratory, Swiss Federal Institute of Technology.

Harney, H. and C. Muckenhirn, 1997. Group Key Management Protocol (GKMP) Architecture, RFC 2094.

Judge, P.Q. and M.H. Ammar, 2002a. Gothic: Group Access Control Architecture For secure Multicast and Anycast. in Proceeding of the IEEE INFOCOM'02, pp: 1547-1556.

Judge, P.Q. and M.H. Ammar, 2000b. WHIM: Watermarking multicast video with a hierarchy of intermediaries. Proc. NOSSDAV, Chapel Hill, NC, pp: 699-712.

Judge P.Q. and M.H. Ammar, 2003. Security issues and solution in multicast content distribution: A Survey. IEEE Network, 17: 30-36.

Mittra, S., 1997. Iolus: A Framework for scalable secure multicasting, In proceedings of ACM SIGCOMM'97, Cannes, France.

Perring, A., R. Canetti, D. Song and J.D. Tygar 2001. Efficient and secure source authentication for multicast, ISOC Network and Distributed System Security Symposium.

Steiner, M., G. Tsudik and M. Waidner. Cliques, 1998. A new approach to group key agreement. IEEE International Conference on Distributed Computing Systems, pp: 380-387.

Steiner, M., G. Tsudik and M. Waidner, 2000. Key agreement in dynamic peer groups, IEEE Transac. Parallel Distrib. Sys., 11: 769-780.

Wallner, D., E. Harder and R. Agee, 1999. Key management for multicast, Issues and Architecture, RFC 2627.

Wong, C., M. Ghonda and S. Lam, 1997. Secure group communications using key graphs. Technical Report TR97-23, University of texas at Austin, Department of computer sciences.

Yacine, C., A. Bouabdallah and H. Seba, 2005. A Taxonomy of Group Key Management Protocols: Issues and Solutions, Transac. Eng. Comput. Technol., 6: 5-17.