Journal of Theoretical and Applied Information Technology
20th August 2014. Vol. 66 No.2

ISSN: 1992-8645 www.jatit.org E-ISSN: 1817-3195

PROFICIENT KEY TREE STRUCTURE FOR DYNAMIC MULTICAST GROUPS

[1]USHA DEVI G, [2]WAHIDHA BANU RSD

[1]School of Information Technology and Engineering, VIT University, Vellore, Tamil Nadu, India

[2]Government College of Engineering, Salem

[1]ushaadevi_g@yahoo.co.in, [2]drwahidabanu@gmail.com

ABSTRACT

Most of the group key management schemes focus on reducing the update messages. Majority of the groups are dynamic in nature and large. To assuage the scalability problem, various key structures have been proposed. In dynamic multicast groups, key updates are complicated since group members join and leave at any time. Hierarchical tree structure is suitable for maintaining multicast group members and keys. If the member wishes to join, it sends request to the Group Centre (GC) or Key Server. The Server authenticates the member and assigns an inimitable ID, which is later inserted into the tree. Similarly, the Server adjusts the tree when the member leaves from the group. Forward and backward secrecy are ensured by updating the keys whenever a join/leave request is handled called re-keying. Proficient Key Tree structure is proposed to reduce the number of re-keying operations during join/leave operations. This tree structure has two parts namely the upper part called quad tree at few levels which has at most four children and the lower part called a binary tree which has at most two children. With this structure, experimental results show that the costs are reduced compared to the full binary key tree approach.

Keywords: Proficient Key Tree Structure, Re-keying, Quad level, Multicast Groups.

1. INTRODUCTION

Many services of multimedia such as pay TV systems and video conferencing are based on multicast group communication where a group of members receive multimedia messages [1]. Groups can be of two types: open or closed. Only registered members can send a message in closed group communications. In contrast, any sender can send a message to group members in an open group. Groups are also categorized into static and dynamic. Joining and leaving from the group is predetermined in static groups and does not change during the communication whereas in dynamic groups, members join/leave the service at any time [2-5].

Secure multimedia communication requires forward and backward secrecy of data [6-9]. The Group Centre (GC) is responsible for disallowing new members to have access to previous data called backward secrecy and disallowing existing members who have left the group to have further access to data called forward secrecy. Forward and backward secrecy are ensured by updating the keys whenever a join/leave request is handled. This process is called as re-keying [10-11]. The number of computations done during the re-keying process called computation cost, the amount of information to be stored called storage cost by the group members and GC [12-13] and the updated keys to be distributed to the other members called communication cost should also be minimized.

Secure multicast services are required for restricting membership of groups since there is a demand in such kind of applications. The major problems of secure communication protocols are scalability [14], the computational complexity of keys, re-keying complexity and the addressing of dynamism in groups [15-16]. The overall objective of this paper is to reduce costs during re-key operations by using proficient key tree structure.

2. RELATED WORKS

Secure multicast group uses binary tree structure to maintain keys [17-22]. The binary tree with 3-levels which maintains keys at all levels as shown in Figure 1. The height of the tree is based on the number of levels in the binary tree and the height H

Journal of Theoretical and Applied Information Technology
20th August 2014. Vol. 66 No.2

ISSN: 1992-8645 www.jatit.org E-ISSN: 1817-3195

is 3 since there are three levels in the binary tree. Multicast group members are inserted into the tree only at leaf level. If 'n' is the number of tree levels, the number of nodes in the tree is 2^{n+1}-1 and the group members are 2^n. The Key tree uses three kinds of keys such as Group Key (GK), Subgroup Key (SK) and an Individual Key (IK). Top level key is a Group key called GK, intermediate level keys are subgroup keys called SKs and leaf level keys are IKs. From the figure, M1 to M8 are members of the multicast group. K0 is the GK, K1 to K6 are SKs and IK1 to IK8 are member's private keys.

The Key Centre distributes new keys to the group members by encrypting them with older keys during the re-keying process. Then, a group member has to decrypt encrypted keys with their old keys. All these encryption and decryption computations increase the load on the KC, resulting in delay while getting the group key. Moreover, it also escalates the power consumption to obtain the new group key. This work proposes a proficient key tree structure by considering the efficiencies of various cost factors.

3. PROFICIENT KEY TREE STRUCTURE (PKS)

Proficient key tree structure has two parts in which the lower part constitutes of binary tree to minimise communication cost and the upper part introduces quad levels to alleviate the computation and storage costs.

The proficient key tree with 1 quad level and 1 binary level for 8 members in a multicast group are shown in Figure 2. From the binary key tree, one GK, 6 SKs and 8 IKs are stored for 8 members at three levels. Compared to the 3-level binary key tree, l-quad level PKS maintains all 8 members at two levels itself and it requires 1 GK, 4 SKs and 8 IKs. The number of SKs can be reduced if quad tree is used. This in turn will reduce storage and computation costs.

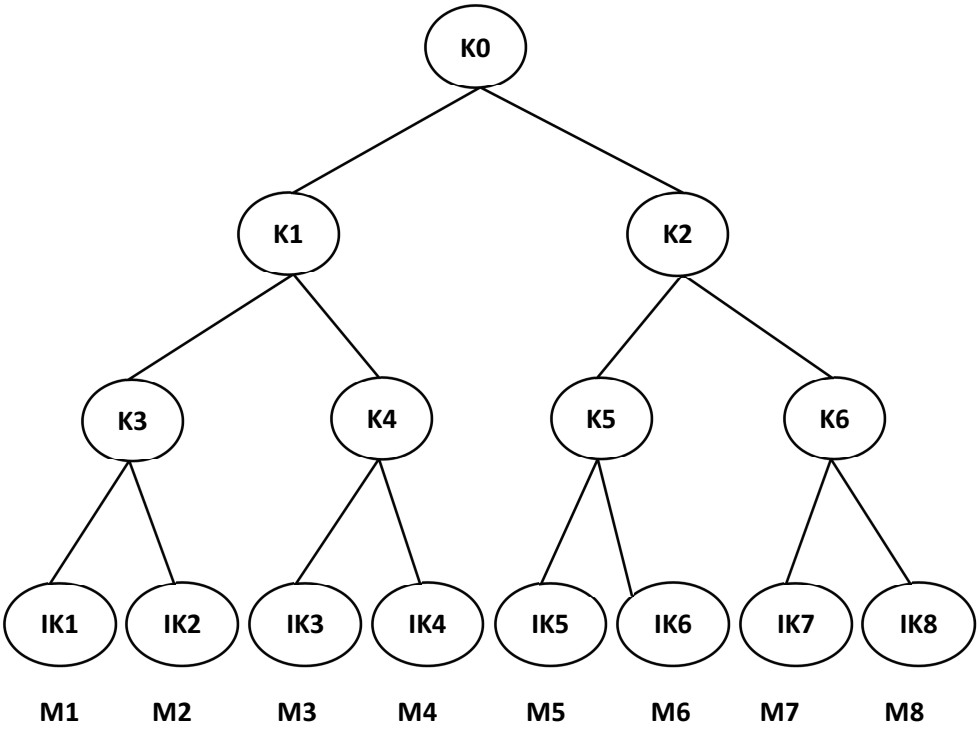

Figure 1. 3-Level Binary Key Tree

Journal of Theoretical and Applied Information Technology
20th August 2014. Vol. 66 No.2

ISSN: 1992-8645 www.jatit.org E-ISSN: 1817-3195

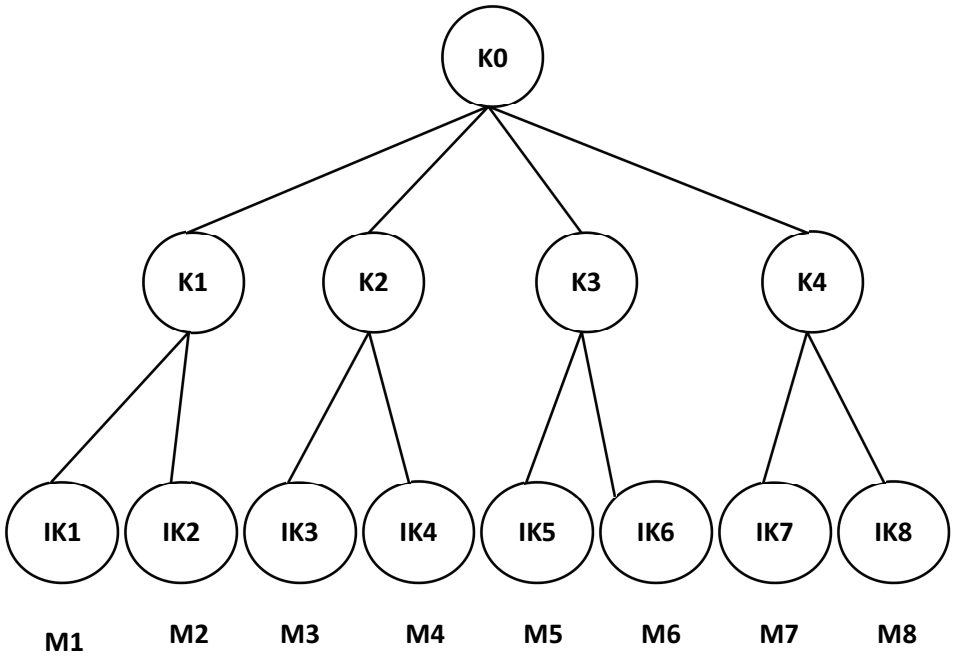

Figure 2. Proficient Key Tree Structure With 1 Quad Level

The proficient key tree with 2 quad and 1 binary levels is shown in Figure 3. Joining and Leaving are the two important operations of a multicast group.

If the member M24 leaves from the group, some key updates are required for maintaining confidentiality and integrity of data. The keys K0, K3 and K16 should be updated as a part of rekeying process. The Key Centre selects a key for encryption of the new key and this new key must be kept secret from the leaving member. This ensures the property of forward secrecy. The number of re-key messages to be signed is reduced through the batch re-keying process which improves efficiency.

4. JOIN OPERATION

When a member (Ui) wishes to join the group, the member has to send a join request. In addition to that, the Key Centre requires new level K' to insert the member into the tree. If the performance of the joining member is not good, then the new level K' is smaller than the present level of the key tree structure (K).

The following steps define an algorithm to join the member in the multicast group.

Step 1: Receive join request from the new member to the multicast group

Step 2: Determine the level of the current tree K and new level required K'

Step 3: If K' is greater than or equal to K, check whether the tree is complete

Step 4: If the tree is a complete tree, remove keys that are on the same level K

Step 5: If K' is less than K, set K as K' and check whether the tree is complete.

Step 6: If the tree is complete, remove keys that are on higher level than or same level as K' else remove keys that are on higher level than K'.

For example, 8 members are at the leaf level and height of the tree K of the lower part is 2 in Figure 1. If the member M9 joins the tree, it is inserted at the leaf level. It requires one more level since the tree is complete. Therefore, the new level K' is 3. In order to make the tree into a PKS tree, quad level is to be introduced. Therefore, the existing key tree structure is to be updated. K1 and K2 are removed and the top level node is directly connected to K3, K4, K5 and K6 nodes after adding the new member at the leaf level.

Journal of Theoretical and Applied Information Technology
20th August 2014. Vol. 66 No.2

ISSN: 1992-8645 www.jatit.org E-ISSN: 1817-3195

5. LEAVE OPERATION

If a group member leaves, a request is sent to the Key Centre. After receiving the request from the member(s), KC has to update the subgroup keys to maintain confidentiality and secrecy of communication. After updating the group keys, it may be a case that the height of the tree is changed due to the empty positions created as a result of leaving members. In such scenarios, height is recalculated for the tree structure's lower part i.e. K". In addition to this, if a group member leaves the group then the height of the tree is updated. Thus, a newly required level K' is determined by considering the memory space and computation power of the remaining members.

The following steps define an algorithm to leave the member in the multicast group.

Step 1: Receive a leave request from the member

Step 2: Update new SKs

Step 3: Recalculate the height of the changed tree K"

Step 4: Determine a new required level K'

Step 5: If both K" and K' are not equal and heightening the level of the tree, create levels from K"+1 to K'.

For example, the member M9 wants to leave the multicast group, K is 2 and the height of the lower part of the changed key tree K" is 1. If K" is smaller than K, the KC increases the height of the tree and generates new SKs on level two.

6. RESULTS AND DISCUSSION

Performance of PKS tree structure is evaluated in three different aspects of cost such as communication cost, computation cost and storage cost. Quad level structure is used for reducing the computation and storage costs. From the binary key tree, one GK, 6 SKs and 8 IKs are stored for 8 members at three levels. Compared to the 3-level binary key tree, l-quad level PKS maintains all 8 members at 2 levels and it requires 1 GK, 4 SKs and 8IKs. The number of SKs is reduced if quad tree is used. This reduces storage and computation costs. In case of 2 quad levels and 1 binary level, 1 SK, 16 SKs and 32 IKs are maintained for 32 members at 3 levels. Storage and Computation costs of PKS tree with 1 and 2 levels of quad structure and binary tree are shown in Table 1. Here N represents the number of members in the multicast group.

Table 1. Storage and computation costs of PKS and binary tree

N	SKs		
	Binary	1-quad level tree	2-quad level tree
32	30	28	20
64	62	60	52
128	126	124	116
256	254	252	244

Table 2. Communication Cost of PKS and Binary Tree

N	Binary tree			1-quad level tree				2-quad level tree			
	H	if e=2	if e=3	K	H	if	if e=3	K	H	if	if e=3
32	5	6.1612	7.7612	3	4	4.645	5.987	1	3	2.967	0.1935
64	6	8.0952	10.589	4	5	6.587	8.8279	2	4	4.936	5.811
128	7	10.055	13.483	5	6	8.551	11.727	3	5	6.913	8.7431
256	8	12.031	16.419	6	7	10.52	14.666	4	6	8.898	11.6981

An analysis of average number of updated messages when 'e' members leave is obtained as follows:

$$cc(binary, e) = \sum_{i=1}^{H} \sum_{Ki=bi}^{Bi} \Pr[n\{e, \frac{N}{\prod_{j=1}^{i} 2}, \prod_{j=1}^{i} 2\} = Ki].Ki - e \quad (1)$$

$$cc(PKS, e) = \sum_{i=1}^{H} \sum_{Ki=bi}^{Bi} \Pr[n\{e, \frac{N}{\prod_{j=1}^{i} 2}, \prod_{j=1}^{i} 2\} = Ki].Ki + (2^{H-K} - 1) - e \quad (2)$$

$$\text{Where } \Pr[n\{e, v, w\} = l] = \frac{C_l^v . F(e,l,w)}{C_e^{vw}} \quad (3)$$

$$\text{and } F(e, l, w) = \sum_{k=0}^{l-b} (-1)^k C_k^l C_e^{w(l-k)} \ (b \le l \le B) \quad (4)$$

Equations 1 state the communication cost of the binary tree and the Equation 2 gives the communication cost of the proficient key tree structure.

Journal of Theoretical and Applied Information Technology
20th August 2014. Vol. 66 No.2

ISSN: 1992-8645 www.jatit.org E-ISSN: 1817-3195

Table 3. Notations in the Communication Cost Calculation

Notations	Description
E	Number of leaving members
H	Height of the tree
b, bi	upper bound of (e/w)
B, Bi	min(e,l)
W	w-sized baskets
L	number of non-full baskets
N	number of nodes
Pr	Probability
F	Number of ways

The cost of communication of three different tree structures is given in Table 2. Here, 'H' refers the tree height, 'K' refers number of binary levels and 'e' refers the number of members leaving from multicast group. The notations used in the equations are described in Table 3.

The proposed key tree structure is tested up to two levels of quad tree and the rest of them are binary levels. It is found that the number of members in a tree is more, the cost of calculating the key decreases while leaving of members. The increase in the communication cost and all other costs depends on number of tree levels.

The number of messages for communicating with other members of the tree when joining or leaving is communication cost. After calculating the communication cost for K values, it is found that the cost is low when compared to the existing model. If the number of levels increases, the cost increases in a binary tree. But in the proposed tree, the cost reduces up to certain levels.

7. CONCLUSION

Reducing the rekeying process by improving the efficiency in terms of storage, computational and communication costs is proposed in this work. In order to reduce costs, the proficient key tree structure is introduced which has two parts in which the lower part constitutes of binary tree to minimise computation cost and the upper part has quad structure up to 2 levels to alleviate communication and storage costs. When the number of levels increases in the lower part of the tree, the communication cost increases. The computation cost and storage cost are also varied based on the number of quad levels. It is compared with binary key tree and the results show that the combination of quad and binary tree gives a considerable reduction in all aspects of costs.

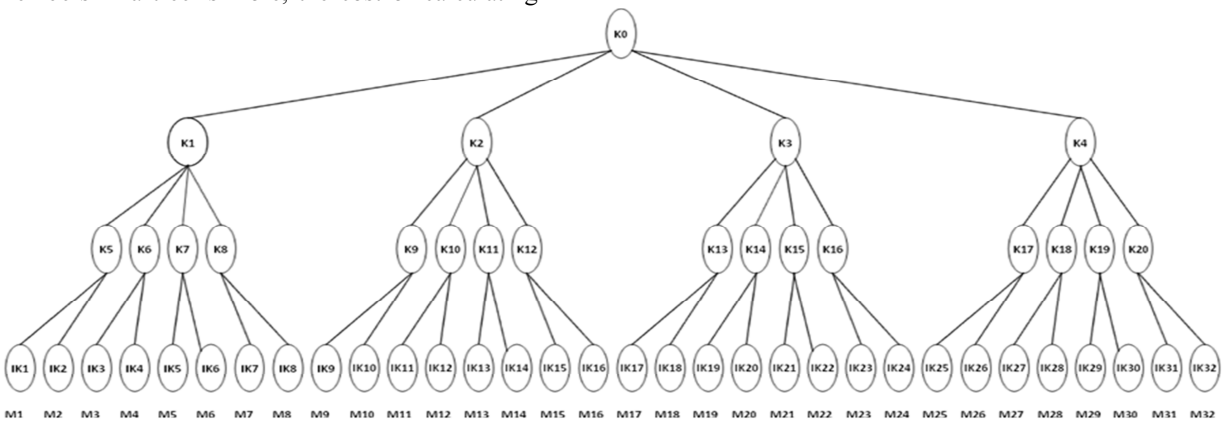

Figure 3. Proficient Key Tree Structure With 2 Quad Levels And 1 Binary Level

ISSN: 1992-8645 www.jatit.org E-ISSN: 1817-3195

REFERENCES:

[1] Wade Trappe, Jie Song, Radha Poovendran, Ray Liu, K.J. "Key management and distribution for secure multimedia multicast", IEEE Transactions on Multimedia, Vol.5 (4), pp.544–557, 2003.

[2] Hock Desmond Ng W, Howarth M, Sun Z, Cruickshank H, "Dynamic Balanced Key Tree management for Secure Multicast Communications", IEEE Transactions on Computers, vol. 56, pp. 590–605, 2007.

[3] Boyd, C. and Mathuria, A. "Key Establishment Protocols for Secure Mobile Communications: A Selective Survey", Information Security and Privacy, LNCS 1438, Springer-Verlag, pp.344-355, 1998.

[4] Molva, R. and Pannetrat, A. "Scalable multicast security with dynamic recipient groups", ACM Transactions on Information and System Security, Vol.3 (3), pp.136–160, 2003.

[5] Sherman, A. T. and McGrew, D. A. "Key establishment in large dynamic groups using one–way function trees", IEEE Transactions of Software Engineering, Vol.29(5), pp.444–458, 2003.

[6] Lihao Xu, Cheng Huang, "Computation Efficient Multicast Key Distribution", IEEE Transactions on Parallel and Distributed Systems, vol. 19, pp. 1–10, 2008.

[7] Hua Chu, H. Qiao, L. and Nahrstedt, K. "A secure multicast protocol with copyright protection", ACM SIGCOMM Computer Communications Review, Vol.32, No.2, 2002.

[8] Rafaeli, S. and Hutchison, D. "A survey of key management for secure group communication", ACM Computing Surveys, Vol.35, No.3, pp.309–329, 2003.

[9] Wong, C. K. Gouda, M. Lam, S. S. "Secure Group Communications Using Key Graphs", IEEE/ACM Transactions on Networking, Vol. 8(1), pp.16–30, 2000.

[10] Jin-Hee C, Ing-Ray C, Mohamed E, "On Optimal Batch Rekeying for Secure Group Communications in Wireless Networks", Journal on Wireless Networks, vol. 14, pp. 915–927, 2008.

[11] Lin, I.C. Tang, S. S. Wang, C. M. "Multicast Key Management without Rekeying Processes", The Computer Journal, Vol. 53 (7), pp.939-950, 2010.

[12] Bezawada Bruhadeshwar, Sandeep S Kulkarni, "Balancing Revocation and Storage Trade-offs in Secure Group Communication", IEEE Transactions on Dependable and Secure Computing, vol. 8, pp. 58–73, 2011.

[13] Dong Hyun Je, Jun Sik Lee, Youngsuk Park, Seung Woo Seo, "Computation and Storage Efficient Key Tree Management Protocol for Secure Group Multicast Communications", Computer Communications, vol. 33, pp. 136-148, 2010.

[14] Hock Desmond Ng W, Sun Z, Cruickshank H, "Scalable Balanced Batch Rekeying for Secure Group Communication", Computers and Security, vol. 25, pp. 265–273, 2006.

[15] Bouassida MS, Chrisment I, Festor O, "Mobility Awareness in Group Key Management Protocols within Mobile Ad Hoc Networks", Annals of Telecommunications, vol. 61, pp. 9-10, 2006.

[16] Usha Devi G, Anusha K, Rajyalakshmi GV, "Performance Comparison of Dynamic Multicast Groups based on Mobility Speed", Journal of Theoretical and Applied Information Technology, Vol.55, No.2, pg.248-253, 2013.

[17] Huang D, Medhi D, "A Secure Group Key Management Scheme for Hierarchical Mobile Ad-Hoc Networks", Ad Hoc Networks, vol. 6, pp. 560-577, 2008.

[18] Lee, J.S. Son, J.H. Park, Y.H. Seo, S.W. "Optimal level-homogeneous tree structure for logical key hierarchy", IEEE Conference on Communication System Software and Middleware Workshop (COMSWARE), 2008.

[19] Lu, H. "A novel high–order tree for secure multicast key management", IEEE Trans. Computers, Vol.54 (2), pp.214–224, 2005.

[20] Sheu, P.-R. Chen, S.T. "A fast and efficient heuristic algorithm for the delay- and delay variation-bounded multicast tree problem", Computer Communications, Vol.25 (8), pp.825–833, 2002.

[21] Antonio Pinto, Manuel Ricardo, "Secure multicast in IPTV services", Computer Networks, Vol.54, pp.1531-1542, 2010.

[22] Usha Devi G., Meesala Sai Kishore, Jyothi Kochuraghavan and Rajakumari M, "Effective Rekeying Mechanism using Minimised Key Tree (MKT) Structure", International Journal of Advanced Research in Computer Science and Software Engineering, Vol.3, No.8, August 2013.