International Journal of Recent Technology and Engineering (IJRTE)
ISSN: 2277-3878, Volume-8 Issue-3, September 2019

Group-Key Generation through Ternary Tree Based Rebuild Algorithm

V.Srinadh, P.V. Nageswara Rao

Abstract: ***Security plays essential role in any correspondence framework particularly in Group oriented correspondence. In Group oriented correspondence, entire correspondence will occur with the help of one secret key which is called Group Key. This gathering key must be changed at whatever point another part joins into the gathering just as a current part leaves from the gathering; which is known as rekeying. This gathering correspondence can be spoken to utilizing key tree. In case we utilize ternary tree, tallness of the tree will be expanded if the group members are more, so that rekeying activity takes additional time. Rather than ternary tree on the off chance that we use quad tree, so that the tree stature will be less, so that rekeying activity takes less time. So the correspondence will be increasingly secure and quicker. In this paper we are going to implement Ternary tree based Rebuild Algorithm.***

Key-Words: - Group key, Rekeying, security, Ternary tree, Rebuild Algorithm

Key-Words: - Group key, Rekeying, backward confidentiality, forward confidentiality, security, Ternary tree

I. INTRODUCTION

Security assumes imperative job in any correspondence framework particularly in Group situated correspondence. In gathering focused correspondence framework whole correspondence will happen with the assistance of one mystery key which is called Group Key. This gathering key must be changed at whatever point another part joins into the gathering. With the goal that the joined individuals can't get to the past imparted information. Just as gathering key must be adjusted at whatever point a current part leaves from the gathering with the goal that the left individuals can't get to the further correspondence information. Changing the gathering key at whatever point another part joins into the gathering and a current part leaves from the gathering is known as rekeying. This gathering correspondence can be spoken to utilizing key tree.

Manuscript published on 30 September 2019
* Correspondence Author
V.Srinadh, P.V*., Assistant professor of Computer Science and Engineering at GMRIT, Rajam, Andhra Pradesh.
Nageswara Rao, Professor, Department of CSE, GITAM University, Visakhapatnam.

On the off chance that we utilize Ternary tree, stature of the tree will be expanded if the individuals are more, so that rekeying activity takes additional time. Rather than Ternary tree on the off chance that we use Quad tree the tree tallness will be less, so that rekeying activity takes less time. To accomplish this we need a protected dispersed gathering key understanding and validation convention with the goal that individuals can set up and confirm a typical gathering key for secure and private correspondence. First key understanding convention was proposed by Diffie-Hellman. It can ensure the security of correspondence between the two clients. Tree-based Group Diffie-Hellman (TGDH) is one of the conventions that stretch out the Diffie-Hellman convention to a gathering key understanding protocol [1].

II. TREE-BASED GROUP DIFFIE–HELLMAN PROTOCOL

To efficiently maintain the group key in a dynamic peer group with more than two members, we use the tree-based group Diffie–Hellman (TGDH) protocol proposed in [10].Each member maintains a set of keys, which are arranged in a hierarchical binary tree. We assign a node ID v to every tree node. For a given node v, we associate a secret (or private) Key Kv and a blinded (or public) key BKv . All arithmetic operations are performed in a cyclic group of prime order p with the generator α . Therefore, the blinded key of node can be generated by

$$BK_v=\alpha^{K_v} \bmod p \quad \text{..... (1)}$$

Each leaf node in the tree corresponds to the individual secret and blinded keys of a group member M_i . Every member holds all the secret keys along its key path starting from its associated leaf node up to the root node. Therefore, the secret key held by the root node is shared by all the members and is regarded as the group key[1].
The node ID of the root node is set to 0. Each non leaf node consists of either three child nodes whose node ID's are given by 3v+1,3v+2 and 3v+3 for its left child, middle child and right child respectively or two child nodes whose node ID's are given by 3v+1 and 3v+2 for its left child and middle child respectively . Based on the Diffie–Hellman protocol [3], the secret key of a non leaf node can be generated, in two cases, as follows

case 1: Non leaf node with two child nodes

Let v be the non leaf node whose two children are 3v+1 and 3v+2 then the secret key of v can be calculated by the secret key of one child node of v and blinded key of another child node of v. Mathematically, we have

$$K_v = (BK_{3v+1})^{K_{3v+2}} \bmod P$$
$$= (BK_{3v+2})^{K_{3v+1}} \bmod P \quad \ldots\ldots\ldots (2)$$
$$= \alpha^{K_{3v+1}K_{3v+2}} \bmod P$$

case 2: Non leaf node with three child nodes

Let v be the non leaf node whose three children are 3v+1, 3v+2 and 3v+2ck then the secret key of v can be calculated by the secret key of one child node of v and blinded key of other two child nodes of v. Mathematically, we have

$$K_v = (BK_{3v+3})^{((BK_{3v+2})^{K_{3v+1}} \bmod P)} \bmod P$$
OR
$$K_v = (BK_{3v+2})^{((BK_{3v+3})^{K_{3v+1}} \bmod P)} \bmod P$$
OR
$$K_v = (BK_{3v+1})^{((BK_{3v+3})^{K_{3v+2}} \bmod P)} \bmod P$$
OR
$$K_v = (BK_{3v+3})^{((BK_{3v+1})^{K_{3v+2}} \bmod P)} \bmod P$$
OR
$$K_v = (BK_{3v+2})^{((BK_{3v+1})^{K_{3v+3}} \bmod P)} \bmod P$$
OR
$$K_v = (BK_{3v+1})^{((BK_{3v+2})^{K_{3v+3}} \bmod P)} \bmod P$$

$\ldots\ldots\ldots (3)$

Fig.1. represents a key tree, which is represented by using ternary tree [2], of a group with 9 members.

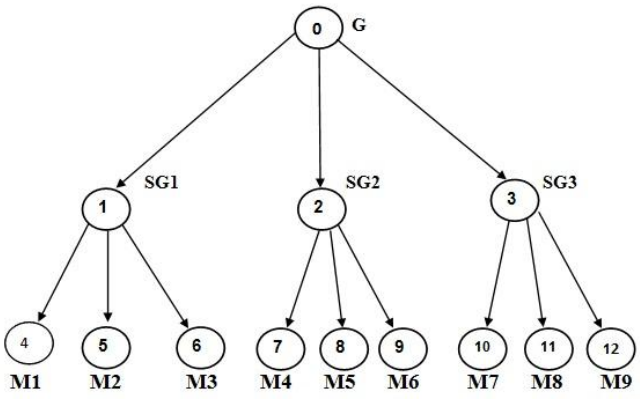

Fig.1. Key tree for a group with 9 members

As the blinded keys are publicly known, every member can compute the keys along its key path to the root node based on its secret key by using the equation (2) and (3). For example, M1 generates the secret key K_4 and it can request the blinded key BK_5 from M2 and BK_6 from M3. M1 generates the secret key K_1 in two ways by using equation (3): either "first use the blinded key BK_5 from M2 and generates new secret key and with this key and blinded key BK_6 from M3, M1 can generate secret key K1" or "first use the blinded key BK_6 from M3 and generates new secret key and with this key and blinded key BK_5 from M2, M1 can generate secret key K1". This secret key K_1 is considered as subgroup key SG1. With the help of blinded keys BK_2 and BK3 secret key K_0 can be computed by using the same procedure. This secret key K0 is said to be group key. Communication will be done using this group key until a new member joins into the group or existing member leaves the group.

Forward confidentiality: Whenever a group member leaves the group then again group key will be changed, so that the left member cannot access further communication data. This is called forward confidentiality [1].

Backward confidentiality: Whenever a group member joins the group then again group key will be changed, so that the joined member cannot access previous communication data. This is called backward confidentiality [1].

III. TERNARY TREE BASED REBUILD ALGORITHM

In singular re-keying, the rekeying is improved the situation each join or leave task. On the off chance that various joins and leaves are performed at the same time at that point figuring rekeying utilizing individual rekeying takes additional time. Rather than individual rekeying consider the interim based circulated re-keying calculations for rekeying, to be specific "The Rebuild algorithm. Intuition: Minimize the final ternary tree height so that the number of rekeying operations of every member is reduced. Basic Idea: Reconstruct the whole ternary tree based key tree to form a complete tree. The fundamental point of the Rebuild calculation is to diminish the tallness of obtained tree. On the off chance that the stature is lessened the rekeying estimations for each group part can be decreased. In the starting of each rekeying interim, the entire key tree will be recreated with every single live member who stays in the correspondence group alongside the recently joining individuals. At the beginning of each rekeying interim if the sum of current members stay in the communication group and the members who are recently joining the group is equal to two then the key tree will be built utilizing Rebuild calculation as appeared in fig.2.

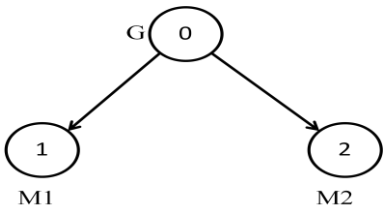

Fig.2. Key tree with two members in Rebuild

At the beginning of each rekeying interim if the sum of current members stay in the communication group and the members who are recently joining the group is equal to three then the key tree will be built utilizing Rebuild calculation as appeared in fig.3.

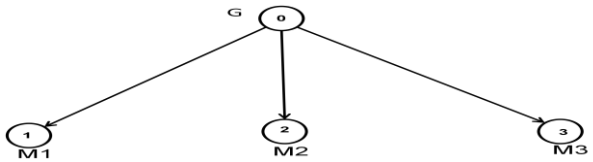

Fig.3. Key tree with three members in Rebuild

At the beginning of each rekeying interim if the sum of current members stay in the communication group and the members who are recently joining the group is equal to four then the key tree will be constructed using Rebuild algorithm as shown in fig.4. If this number is equal to five then the key tree will be constructed using Rebuild algorithm as shown in fig.5. If this number is equal to six then the key tree will be constructed using Rebuild algorithm as shown in fig.6.

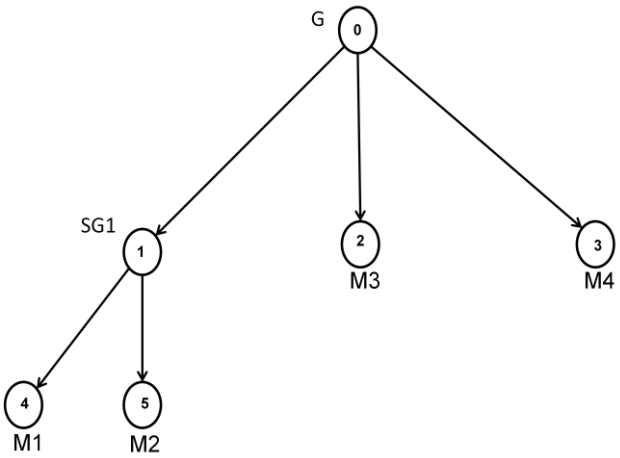

Fig.4. Key tree with four members in Rebuild

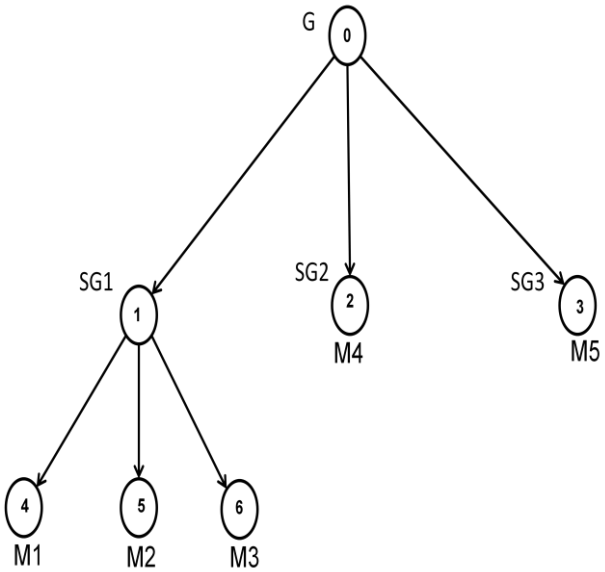

Fig.5. Key tree with five members in Rebuild

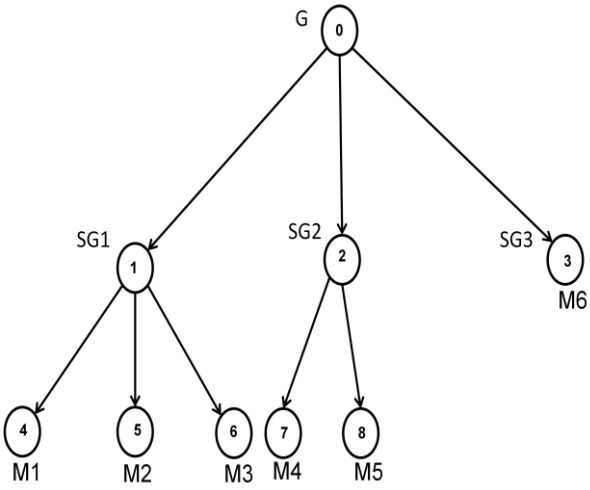

Fig.6.Key tree with six members in Rebuild

IV. REKEYING

Rekeying (renewing the keys of the nodes) is performed for every single (multiple) join / leave event to ensure backward and forward confidentiality. A special member called sponsor is elected to be responsible for broadcasting updated blinded keys. Consider the key tree with 12 members (M1 to M12) as shown in figure 7. Consider the scenario If M2, M4, M5, M7, M9 and M12 leave the group while M13, M14, M15 and M16 join the group. M8 joins the group. Then Ternary tree based rebuild algorithm works as follows: First, identify the list of members available in the group.Here the group is having 12 members M1 to M12. Unlike individual rekeying, Ternary tree based Rebuild Algorithm will perform rekeying one time per one set of leaves and joins simultaneously. Now remove the leaving nodes i.e. remove M2, M4, M5, M7, M9 and M12 from the list and add the joining nodes M13, M14, M15 and M16 to the list simultaneously. So the new list is with group members M1, M3, M6, M8, M10, M11, M13, M14, M15 and M16

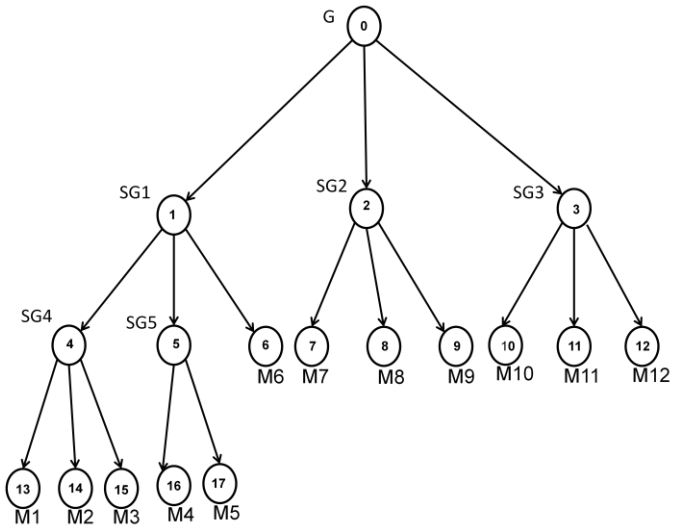

Fig.7. Key tree with 12 members

As the new list is having 10 members the Ternary tree based Rebuild algorithm constructs the key tree as shown in fig.8. Now the group key has to be changed. The rekeying operations: M13, M14, M15, and M16 broadcast their blinded keys on their joining.M16 is one of the sponsors.M16 computes K3 (SG3) using its private key K12 along with M14's blinded key BK10 and M15's blinded key BK11.M16 broadcasts BK3.M13 is one of the sponsors.

M13 computes K2 (SG2) using its private key K9 along with M10's blinded key BK7 and M11's blinded key BK8.M13 broadcasts BK2.M3 will become one of the sponsors. M3 computes K4 (SG4) using its private key K14 and M1's blinded key BK13.M3 broadcasts BK4. M3 computes secret key K1 using K4, M6's blinded key BK5 and M8's blinded key BK6. M3 broadcasts BK1.Using BK1 and BK2, M16 can compute K0 which is group key. The same way remaining members can

compute secure group key. Communication will happen as long as there is no leave event or join event or both. When there is a member wants to join the group or leave the group then the group-key will be changed and a similar way it will be processed.

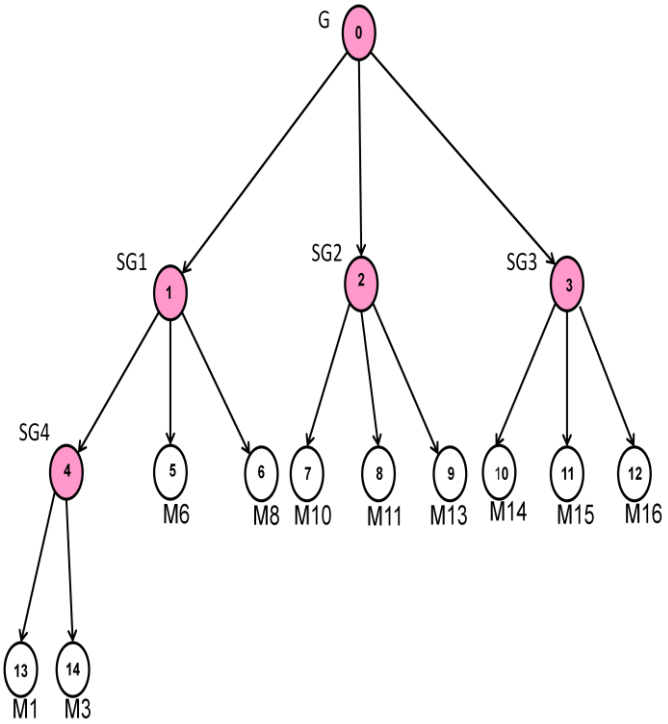

Fig.8. Key tree using Rebuild after M2, M4, M5, M7, M9 and M12 leave the group while M13, M14, M15, and M16 join the group

V. MATHEMATICAL ANALYSIS

Consider a group with N members and Let L be the number of members leaving the group and J be the number of members joining the group. Now the resultant group members N* can be calculated as follows:

$$N^*=N-L+J \qquad \text{..... (4)}$$

Table.1. Height of the binary tree and ternary tree for a different number of members of the group.

No. Of Nodes	Height of the Binary tree	Height of the Binary tree
1	0	0
2	1	0.63
3	1.58	1
4	2	1.26
5	2.32	1.46
6	2.58	1.63
7	2.81	1.77
8	3	1.89
9	3.17	2
10	3.32	2.1
11	3.46	2.18
12	3.58	2.26
13	3.7	2.33
14	3.81	2.4
15	3.91	2.46
16	4	2.52
17	4.09	2.58
18	4.17	2.63
19	4.25	2.68
20	4.32	2.73
21	4.39	2.77
22	4.46	2.81
23	4.52	2.85
24	4.58	2.89
25	4.64	2.93
26	4.7	2.97
27	4.75	3
28	4.81	3.03
29	4.86	3.07
30	4.91	3.1
31	4.95	3.13
32	5	3.15
33	5.04	3.18
34	5.09	3.21
35	5.13	3.24
36	5.17	3.26
37	5.21	3.29
38	5.25	3.31
39	5.29	3.33
40	5.32	3.36
41	5.36	3.38

Height of the binary tree and ternary tree for a different number of members of the group can be represented as shown in the Table 1. The tree height comparison graph is as shown in fig.9.

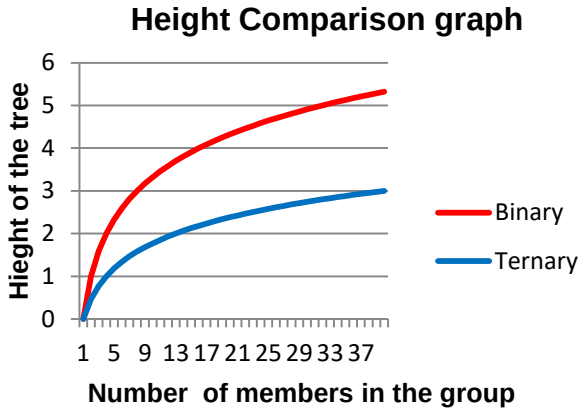

Fig.9. Comparison graph between Binary Rebuild and Ternary Rebuild

VI. CONCLUSION

To represent group members if we use binary tree, height of the tree will be increased if the members are more, so that rekeying operation takes more time. Instead of Binary tree if we use ternary tree the tree height is less compared to binary tree, so that rekeying operation takes less time. So the communication will be more secure and faster.

REFERENCES:

1. Patrick P. C. Lee, John C. S. Lui and David K. Y. Yau "Distributed Collaborative Key Agreement and Authentication Protocols for Dynamic Peer Groups" IEEE/ACM transactions on Networking, VOL. 14, NO. 2, April,2006
2. N.Renuga devi, C.Mala, "Ternary Tree Based
3. Group Key Agreement for Cognitive Radio MANETS" I.J. Computer Network and Information Security, 2014, 10, 24-31.
4. W. Diffie and M. Hellman, "New directions in cryptography," IEEETrans. Inf. Theory, vol. 22, no. 6, pp. 644–654, 1976.
5. T. Sherman and D. A. McGrew, "Key establishment in large dynamic groups using one-way function trees," IEEE Trans. Software Eng., vol. 29, no. 5, pp. 444–458, May 2003.
6. M. Steiner, G. Tsudik, and M. Waidner, "Key agreement in dynamic peer groups," IEEE Trans. Parallel Distrib. Syst., vol. 11, no. 8, pp.769–780, Aug. 2000.
7. M. Waldvogel, G. Caronni, D. Sun, N. Weiler, and B. Plattner, "The versakey framework: versatile group key management," IEEE J. Sel. Areas Commun., vol. 17, no. 9, pp. 1614–1631, Sep. 1999.
8. K.Wong, M. Gouda, and S. S. Lam, "Secure group communications using key graphs," IEEE/ACM Trans. Netw., vol. 8, no. 1, pp. 16–30,Feb. 2000.
9. B.Song and K.Kim"Two-pass authenticatedkey agreement protocol with key confirmation" in Proc. IndoCrypt, Dec. 2000, vol. LNCS 1977, pp. 237–249.
10. "Tree-based group key agreement," ACMTrans. Inf. Syst. Security, vol. 7, no. 1, pp.
11. 60–96, Feb. 2004.

AUTHORS PROFILE

Srinadh.Vemireddi received the Ph.Degree in Computer Science from the University of GITAM University, Visakhapatnam in March, 2019. He is currently an Assistant professor of Computer Science and Engineering at GMRIT, Rajam, Andhra Pradesh.

P.V. Nageswara Rao received M.Tech (CST) from Andhra University and Ph.D from Nagarjuna University and had 23 years of teaching and research experience. Presently working as Professor, Department of CSE, GITAM University, Visakhapatnam. Research interests include Bioinformatics, Soft Computing and Data Mining.